The Environment Concept Model
Builder's Manual

Prepared By
VisiTech, Ltd.
33107 North 18th Street
Arlington, Virginia 22201
(703) 391-6264

Prepared For
Commander
Naval Research Laboratory
4555 Overlook Ave., SW
Washington, DC 20375
Attention: Advanced Information Technology Branch, NRL Code 5580

Edition 1.0
16 August 1999

Table of Contents

List of Figures

1. Introduction

Because environment representation, object models and object behaviors are so clearly linked, valid virtual prototype behaviors require consistent environment representations. Colloquially expressed, a consistent environment representation means that "everyone plays on the same day". More rigorously, consistent synthetic natural environments provide representations that are valid to a chosen resolution, and are spatially, temporally and spectrally continuous.

Responsive simulations must generate the prototype behaviors of interest, and they must be available in a timely fashion, for the investment that the program or project can afford. "Just enough" environment representations have been implemented within the constraints of the overall implementation schedule and budget.

To achieve simulations where "everyone plays on the same day" with "just enough" environment representation, we make use of three important tools: the Environment Reference Framework, the Environment Reference Implementation Process, and the Environment Concept Model (ECM). Although this manual is workbook for creating ECMs, this section will start by putting the ECM in the larger context of an overall approach to environment representation in simulation.

1.1 The Environment Reference Framework

The Environment Reference Framework[1], (often known as the Birkel diagram), provides a common basis for discussing environment representation in simulation. Figure 1. shows the Environment Reference Framework.

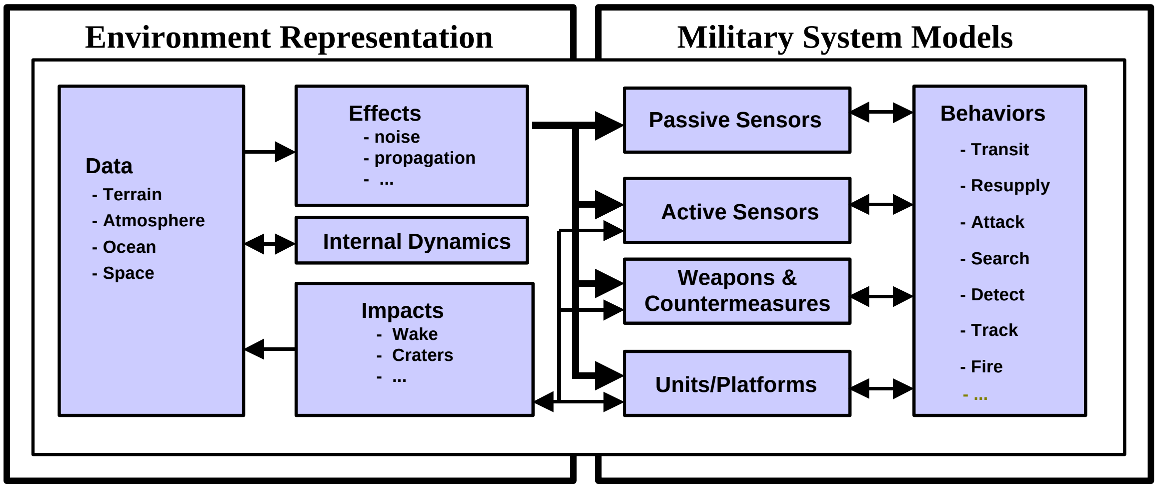


Figure 1 Environment Reference Framework

Environment representations are created from combination of environment data, effects calculations, internal dynamics calculations, and impacts calculations. Environment data can describe many domains: terrain, vegetation, rivers, ocean surfaces, the water column, bathymetry, the surf zone, atmosphere, weather conditions, or even space. This data can consist of sets of direct measurements, parameterized descriptions, or other forms. Effects calculations use data as input to create descriptors that affect modeled object states. Some common calculation examples include propagation loss and scattering of radiated

[1] The Environment Reference Framework is not a prescribed architecture, or a prescribed logical or physical simulation implementation. The framework is the point of departure for simulation professionals to have a common understanding of the nature and scope of environment representation in simulation.

energy, ocean surface wave spectra that drive marine vehicle dynamics, or cloud cover volume distributions that affect intervisibility between aircraft and ground objects. Internal dynamics calculations use data as the basis for calculating changes to the environmental state due to natural causes, e.g. tides, weather fronts, etc. Impacts calculations calculate changes in the environmental state due to the influence of simulation objects, e.g. aircraft generating contrails, ships generating wakes, active sonar pulses raising the background acoustic noise level, etc.

The heart of any simulation is the object representations, and their associated behaviors. The objects respond to environment data and effects, and certain object behaviors impact the environment. From a simulation user's viewpoint, the most important element of the simulation are the object behaviors which occur when the simulation is exercised.

The Environment Reference Framework clearly shows the connection between environment representation and object behaviors, wherever those behaviors are affected by the natural environment. Objects which have behaviors affected by the natural environment must be "environment-aware"; they must accept and change state as a result of environment parameters. The values of those object parameters may be created by accessing both data and calculations, or both. Further, the environment representation, when taken as a whole, must be unitary; clouds must move in response to winds, terrain must be submerged or exposed according to tidal patterns, and so on. The Environment Reference Framework encourages a unified approach to specifying natural environment representations, regardless of where the data and calculations are drawn from, or where they are installed in the implemented hardware/software configuration.

1.2 The Environment Reference Implementation Process

The Environment Reference Implementation Process was developed specifically to support a program simulation system engineer in his effort to deliver responsive simulation capability on time and within budget. The process develops a recommended set of environment data, calculation models and heuristics that constitute “just enough” consistent environment representation to produce valid simulation results. Figure 2 shows the Environment Reference Implementation Process.

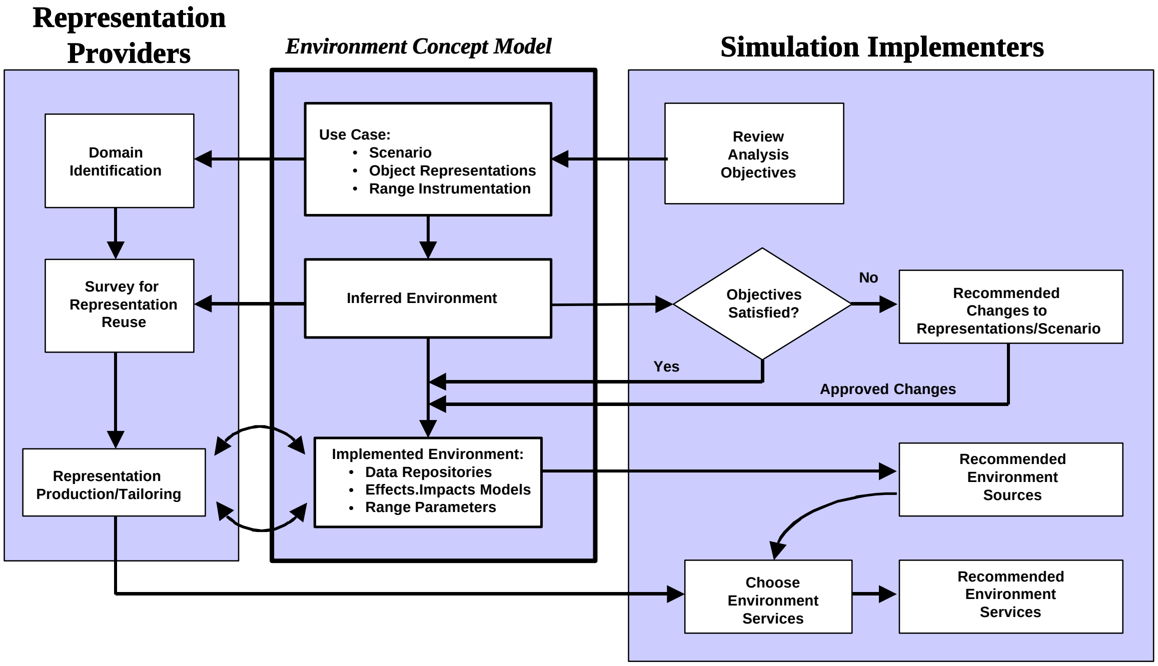


Figure 2 Environment Reference Implementation Process

The process begins by reviewing the underlying operational scenario, and the participants (both real and modeled). If the simulation will be used in support of test range activities, then range instrumentation is reviewed as well. This initial review defines the application use case. Based on this review we can develop a unified environment representation (the inferred environment) that is the logical outgrowth of the objective-based use case. If necessary, we can then make recommendations for changes to scenarios or military systems models, if there is no other cost-effective way to satisfy environment representation requirements.

The process continues by selecting the set of environment data, models and, if applicable, range measurement parameters that constitute the needed environment representation, specific to the simulation requirements at hand (the implemented view). Some simulation implementers prefer a traditional, waterfall sequential approach to simulation implementation. Others prefer an evolutionary spiral implementation. The Environment Reference Implementation Process fits within a variety of approaches, always emphasizing a collaboration between simulation implementations and natural representation providers throughout the implementation life cycle.

The environment selection process is accompanied by a documentation capability, the ECM, that complements sound engineering judgement with standards-based software modeling languages and tools. The ECM describes a unified environment representation (wherever it may occur) and makes it as accessible as every other component of the simulation implementation.

The key point is that the ECM is the outgrowth of a collaborative process that coordinates the efforts of simulation implementers and environment representation providers. We describe the collaboration team as being composed of participants filling four roles: simulation system engineers, simulation developers, environment system engineers, and environment domain experts. We are not suggesting a particular organization; rather, these roles encompass the core activities to be accomplished during the environment implementation process.

We have found that any documentation approach should recognize the need for:

a. An incrementally constructed documentation approach that delivers partial products throughout the simulation implementation process
b. A use case description that describes the context for the simulation application, providing vital context information for the environment representation
c An environment representation description reflecting the use case needs, without the compromises inherent in any implementation
d. A description of the environment representation as it is implemented in the simulation.

1.3 The Environment Concept Model Defined

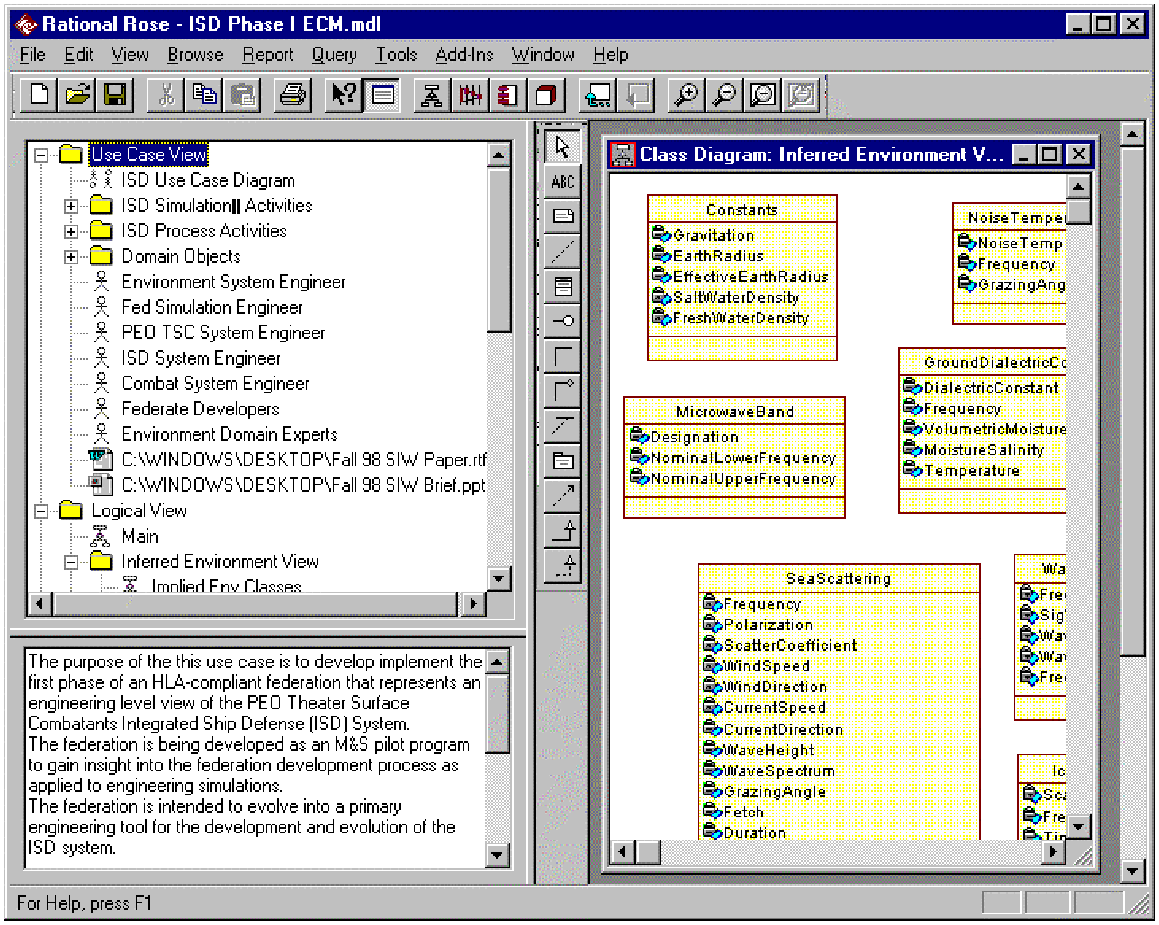
The ECM is a unified description of the synthetic natural environment for a simulation application. The heart of the ECM is an object oriented analysis and design level model, implemented in a standards-based object oriented analysis and design language. The object oriented model may be augmented by referenced electronic documents which amplify aspects of the representation. The object model may also be augmented by referenced files containing specific formatted queries and download requests from environment information repositories.

The core object oriented model describes computational processes and process interactions as well as static data structures. The purpose of the core object model is to unambiguously describe the environment representation to be used in the simulation application. The augmenting electronic documents and files may consist of reports, technical literature, briefings, test condition matrices, or other information that clarifies the reasoning behind selections of particular parameters or algorithms.

Because the ECM is not intended to be implementation-dependant, the ECM and its augments do not contain actual formatted environment data, source/executable code, or heuristics. However, in practical use, we have found that it is important to document allocations of environment representation (specific parameter files or calculation processes) to components of the simulation system. Further, by specifying a standards-based language for the object-oriented model, we anticipate greater future use of automated data schemas and code generation, and code parsing to reverse engineer legacy code. Thus, the boundaries between ECM documentation and simulation content may blur as software engineering practice smoothes the evolution from design to runtime execution.

1.3.1 Object Oriented Modeling

Object oriented modeling describes the physical world in an intuitive representation that can be directly replicated in software. Physical entities (objects) are described by their characteristics (attributes) and their behaviors (operations). The entities may be grouped into sets (classes) in hierarchies (subclasses and superclasses). The physical entities may exchange information (messages). Sometimes an object with its attributes and operations may represent an example of a category of physical entities (a stereotype or a type). Numbers of non-hierarchically related objects may be associated together (packages). When implemented in software, modules of code (components) may be installed (deployed) on one or more processing hosts.



The purpose of the this use case is to develop implement the first phase of an HLA-compliant federation that represents an engineering level view of the PEO Theater Surface Combatants Integrated Ship Defense (ISD) System.
The federation is being developed as an M&S pilot program to gain insight into the federation development process as applied to engineering simulations.
The federation is intended to evolve into a primary engineering tool for the development and evolution of the ISD system.

Figure 3 Rational Rose Display

Object oriented design languages use a descriptive notation that attempts to unambiguously define the emerging representation. Often, languages are both graphical and textual, with diagrams and terms having an agreed-upon grammar and syntax. Different types of symbols and notations are placed in different types of diagrams to describe (expose) the design from different perspectives (views).

Modern object oriented design tools to create sufficiently complete and unambiguous descriptions that the resulting files can be input to code generators to create database designs or source code. Increasingly, design tools have parsers that are able to reverse engineer source code into design diagrams (round trip engineering). Figure 3 on the previous page shows an example of one such modeling tool: Rational Rose, from Rational Corporation.

We suggest two texts that will amplify this skeletal description. The first, Fowler's UML Distilled[2], provides an overview of object oriented analysis and design from several viewpoints as well as introducing the reader to the Unified Modeling Language. The second, Booch/Rumbaugh/Jacobson's The Unified Modeling Language User Guide[3] is a more focused, detailed explanation of the applications and conventions of the Unified Modeling Language (UML).

1.3.2 A Note about Tools

Building and using ECMs doesn't require that one use a particular modeling object oriented language, or document your work using a particular tool. If you wish, you may prefer Rumbaugh's Object Model Technique, Schlaer-Mellor , or the Unified Modeling Language (UML). However, whatever language you use, you should satisfy yourself that it can represent your application's environment representation in a way that communicates effectively with the entire simulation team. Similarly, you may choose from any number of documentation tools. For us a good tool is tolerably easy to use, allows you to reuse all or part of your previously developed ECMs, and integrates into your team's round trip engineering process.
We use the Unified Modeling Language, developed by Rumbaugh, Booch and Jacobson and now being maintained by the Object Management Group.

To document our models, we have used Rational Corporation's Rational Rose. However, over a dozen vendors now offer UML modeling tools with varying features, in a range of prices and licensing conditions. Our experience shows that a suitable tool should have, at minimum, the following features:

a. Follows the notation conventions established for the modeling language
b. Documents requirements and stakeholders explicitly, in addition to the environment representation schema
c. Document dynamic (message passing, changing participants, etc.) as well as static (stakeholders, activities, physical objects to be simulated) aspects of the simulation scenario.
d. Provides features for representing different views of the same simulation
e. Provide a capability to link external files with elements contained within the core object model file(s).
f. Provide graphical views and tools for creating and editing object model views
g. Provide file export and printing of selected object model views

In addition, preferred modeling tools have the following additional features:

a. Exports and imports models views to and from a commonly accessible model repository
b. Generates data schemas, interface definitions, structured queries, and code structures from the object model
c. Parses database schemas or source code into object model content
d. Provides HTML file output of selected object model views
e. Provides automated modeling syntax checking, to avoid notation errors.

Above all, good models create insight and understanding. That's the goal of the ECM, and the selected modeling tool should facilitate that goal.

[2] Fowler, Martin, Scott, Kendall, UML Distilled, Addison Wesley, 1997
[3] Booch, Grady, Rumbaugh, James, Jacobson, Ivar, The Unified Modeling Language User Guide, Addison-Wesley, 1999.

2. Building the Environment Concept Model

Fundamentally, an ECM is composed of three different perspectives on synthetic natural environment representations. One perspective involves documenting the underlying need for the simulation application, and the battlespace to be simulated. A second perspective involves describing the natural environment required to represent the battlespace. A third perspective involves describing the natural environment implemented for the simulation application.

The use case view uses modeling to create perspective and background for the environment representation task. In the use case view is information about the stakeholders who contribute to, or benefit from, the simulation; activities surrounding the simulation; and a description of the battlespace. The use case view also references supporting documentation that helps to frame the requirements for environment representation.

The inferred view is the environment representation that is inferred by the characteristics of the use case. By "inferred", we mean the representation that is a logical extension of the (a) level of fidelity appropriate to the purpose of the use case, (b) the domains and bandwidths in which the participating objects operate, and (c) the space and time described in the scenario.

The implemented view is the environment representation intended for use in the simulation. Ideally, the implemented view would appear very similar to the inferred view. However, we have found that there is often a difference between the environment representation inferred by the use case and the representation that is implemented for the simulation. The system engineer's goal is to deliver a simulation capability that fulfills application needs; environment representation investments are balanced against hardware purchase needs, integration testing needs, etc. Differences can develop because of implementation schedule and cost constraints, lack of suitable data sets, inability to modify proprietary simulation software, to name just a few reasons.

It's often feasible to develop the inferred and implemented views in parallel, and to compare them frequently. The emerging difference between the inferred and implemented representations often provides valuable insight into sources of approximation in the simulation results. If the simulation system engineer is uncomfortable with the impact of the approximations, then there is still time to change the implemented environment to more fully reflect the inferred environment representation. The implemented environment view describes the actual parameters, data sets and models implemented in the simulation.

2.1. Building the Use Case View

In UML, the Use Case View is used to show the intended behavior of the system to be implemented. We develop the Use Case View to fosters a common understanding between the federation system engineer, the federation developer, the environment system engineer, and the environment domain experts about the system being simulated. The Use Case View explicitly describes the participants in the overall project (the actors), the processes to be simulated (the simulation activities), and other relevant project efforts that are not being simulated (the non-simulation activities). Figure 4, on the next page, shows the general layout of a Rational Rose display of the Use Case View.

In the example project we have prepared and shown one set of use case diagrams. If the application is complex, there may be many use case descriptions in the Use Case View, each documenting a discrete subset of the application space. Object-oriented modelers often build models with dozens of use cases to insure that system implementers understand who interacts with the system, and what the system is supposed to accomplish. In the same spirit, if your simulation application is to be used by different communities of users for different purposes (for instance, a training system used for mission rehearsal and, possible, advanced concept exploration), then it's wise to document each community's intent in a separate use case.

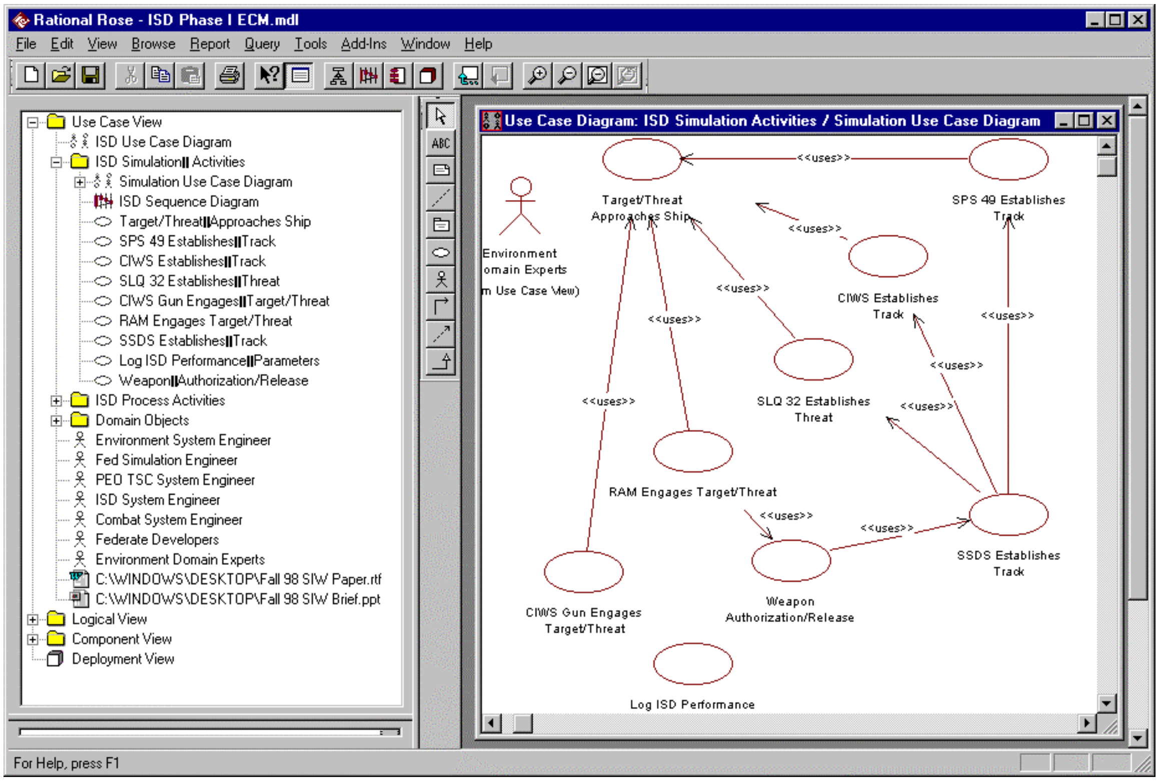

Rational Rose - ISD Phase I ECM.mdl
File Edit View Browse Report Query Tools Add-Ins Window Help
Use Case View
ISD Use Case Diagram
ISD Simulation Activities
Simulation Use Case Diagram
ISD Sequence Diagram
Target/Threat Approaches Ship
SPS 49 Establishes Track
CIWS Establishes Track
SLQ 32 Establishes Threat
CIWS Gun Engages Target/Threat
RAM Engages Target/Threat
SSDS Establishes Track
Log ISD Performance Parameters
Weapon Authorization/Release
ISD Process Activities
Domain Objects
Environment System Engineer
Fed Simulation Engineer
PEO TSC System Engineer
ISD System Engineer
Combat System Engineer
Federate Developers
Environment Domain Experts
C:\WINDOWS\DESKTOP\Fall 98 SIW Paper.rtf
C:\WINDOWS\DESKTOP\Fall 98 SIW Brief.ppt
Logical View
Component View
Deployment View
Use Case Diagram: ISD Simulation Activities / Simulation Use Case Diagram
Environment
omain Experts
m Use Case View)
Target/Threat Approaches Ship
SPS 49 Establishes Track
CIWS Establishes Track
SLQ 32 Establishes Threat
RAM Engages Target/Threat
SSDS Establishes Track
CIWS Gun Engages Target/Threat
Weapon Authorization/Release
Log ISD Performance
<<uses>>
For Help, press F1

Figure 4 Use Case View

2.1.1 Use Case View Step 1: List the Stakeholders (Actors) and Activities

The Use Case diagram records actors, allowing the ECM developers to identify the stakeholders and customers for the ECM early in the environment development process. We also like to list the actors because this is usually an important indicator for the number of participants in developing an environment representation. If there are fewer, more well-connected activities, then descriptions can be courser, and collaboration is easier. If the use case lists a large number of activities, then the ECM becomes an important collaboration tool in defining environment representation… and should be developed accordingly.

In the application example, the non-simulation activities infer the need for a simulation that produces outputs which can be compared with readouts from test instrumentation. Thus, identified non-simulation activities expose important information about the simulation's needed fidelity and accuracy.
Furthermore, non-simulation activities may have associated objects, just as simulation activities have associated objects. For instance, the non-simulation activity "Gather test range weather measurements" might have the associated objects such as "Radiosonde" or "Anemometer". When insitu environment data is collected as part of a test, exercise or battlespace activities, these data are often an important element of project activities. The Use Case Diagram documents these collection activities.

The Use Case Diagram records simulation activities because the activities answer the question "what activities are being modeled/simulated?" The simulation activities are groups of simulation events packaged as a short description. When we say that the ECM is application-specific, we mean that the environment described in the ECM is explicitly engineered to satisfy a known set of requirements… requirements levied by those data and computations needed to execute the packaged simulation events represented by one or more simulation activities. In general wargaming applications or for training, the simulation activity descriptions sound very much like task elements in mission task lists. For detailed

simulations supporting T&E the activity descriptions might be drawn from test events. For concept and performance evaluation applications, the activities might be scenario operational/tactical situations.
In every case, the Use Case simulation activities cite real world activities… with one exception. Every use case diagram has a simulation activity entitled "Log simulation data" as a reminder that logging function must be explicitly considered in developing synthetic environment representations. Figure 5 shows the top level use case diagram and Figure 6 shows a diagram focusing on simulation activities.

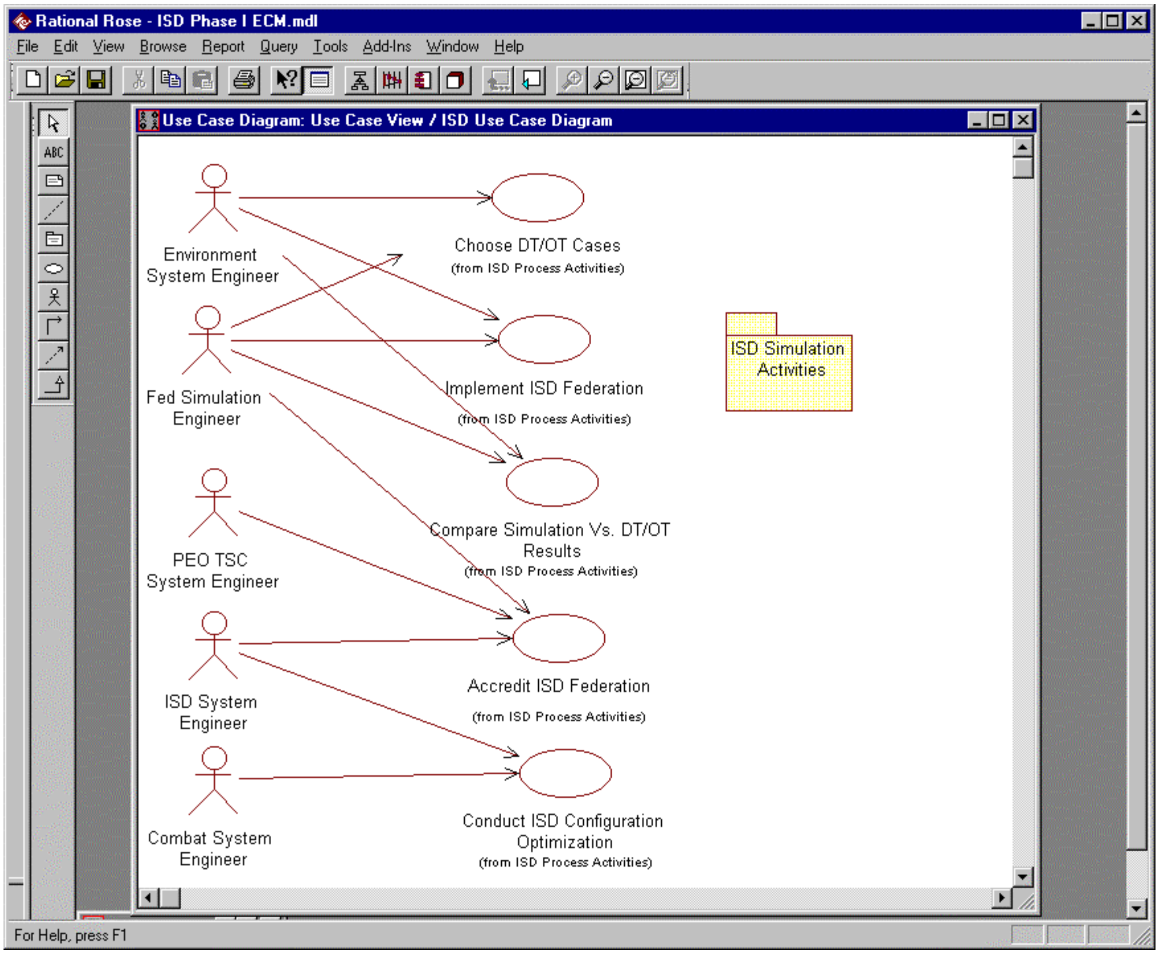

Rational Rose - ISD Phase I ECM.mdl
File Edit View Browse Report Query Tools Add-Ins Window Help
ABC
Use Case Diagram: Use Case View / ISD Use Case Diagram
Environment System Engineer
Fed Simulation Engineer
PEO TSC System Engineer
ISD System Engineer
Combat System Engineer
Choose DT/OT Cases
(from ISD Process Activities)
Implement ISD Federation
(from ISD Process Activities)
Compare Simulation Vs. DT/OT Results
(from ISD Process Activities)
Accredit ISD Federation
(from ISD Process Activities)
Conduct ISD Configuration Optimization
(from ISD Process Activities)
ISD Simulation Activities
For Help, press F1

Figure 5 Top Level Use Case Diagram

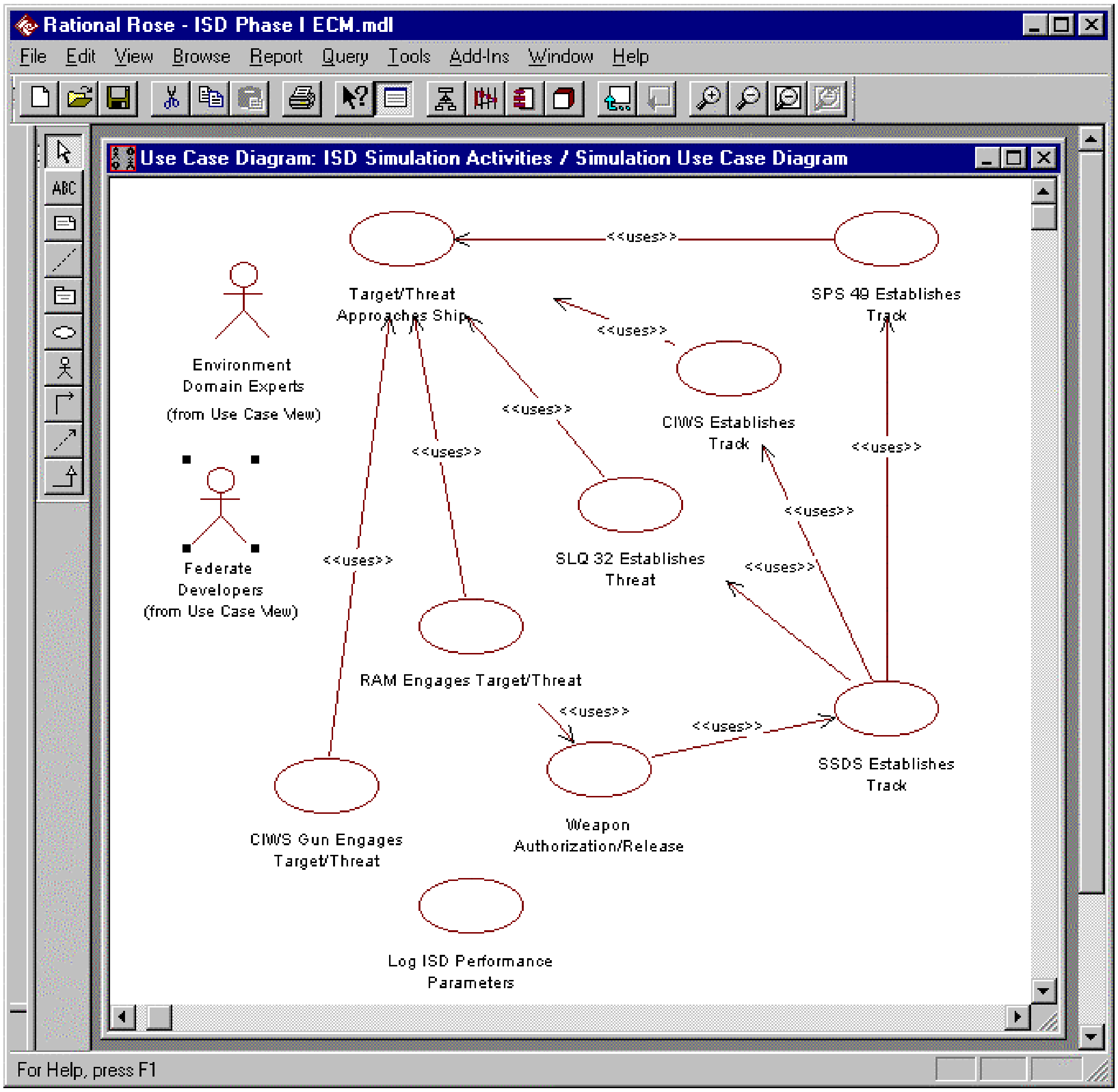

Rational Rose - ISD Phase I ECM.mdl
File Edit View Browse Report Query Tools Add-Ins Window Help
Use Case Diagram: ISD Simulation Activities / Simulation Use Case Diagram
Target/Threat Approaches Ship
SPS 49 Establishes Track
<<uses>>
Environment Domain Experts
(from Use Case View)
CIWS Establishes Track
SLQ 32 Establishes Threat
Federate Developers
(from Use Case View)
RAM Engages Target/Threat
SSDS Establishes Track
Weapon Authorization/Release
CIWS Gun Engages Target/Threat
Log ISD Performance Parameters
For Help, press F1

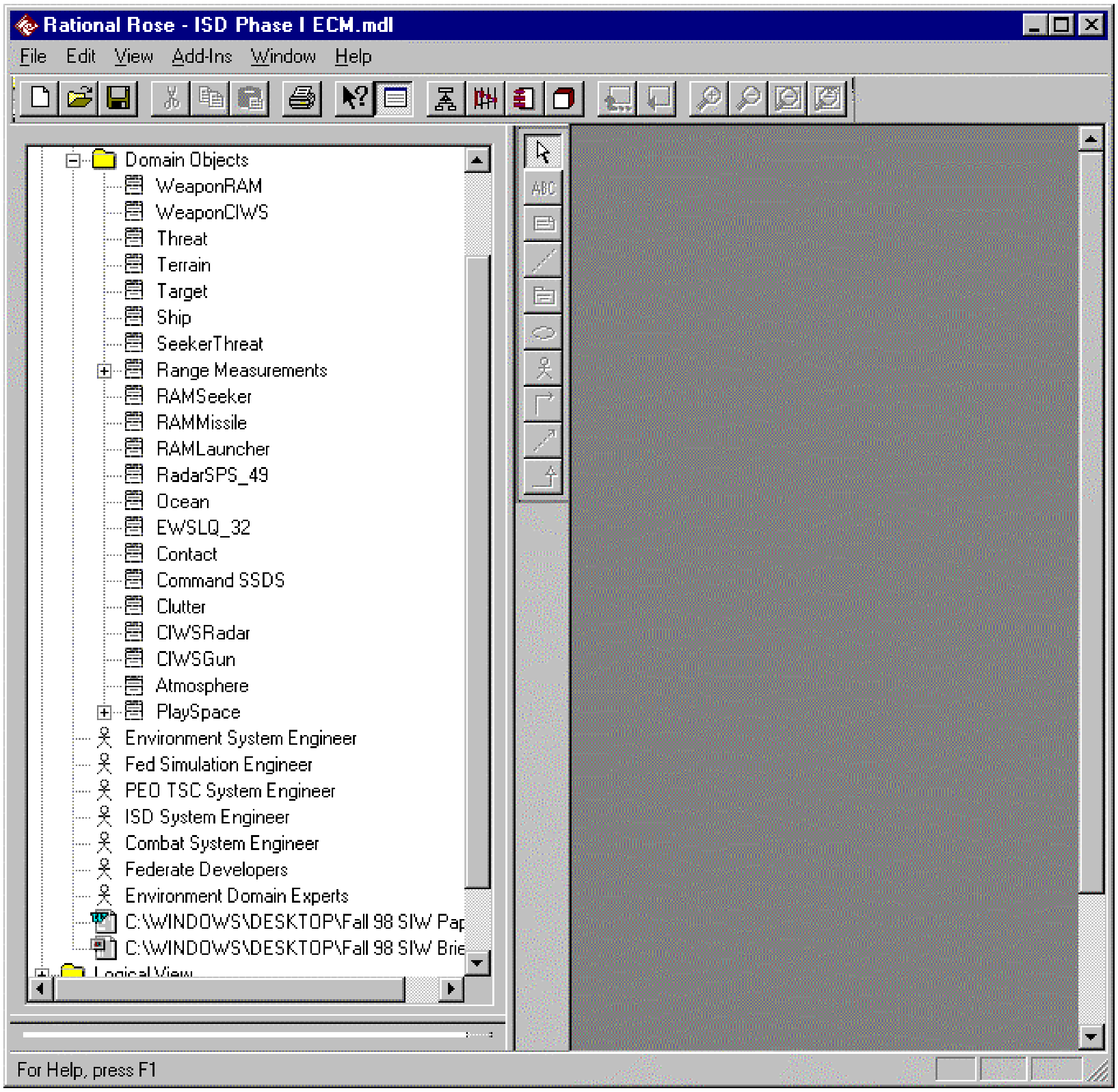
Figure 6 Simulation Activities Use Case Diagram

2.1.2 Use Case View Step 2: List the Simulation Participants (Objects)

The simulation activity descriptions have associated lists of objects. These objects are the participants in the simulation: ships, aircraft, battalions, command structures, etc. Taken as a whole, the objects associated with the simulation activities correspond to the order of battle. The list of objects might be assembled from archived simulation set-up files, from warfare tactical databases, from exercise asset lists, etc. The main object here is to describe the objects in sufficient detail to identify the environment-sensitive objects, e.g. Platforms, communications systems, sensor systems, weapon and weapon guidance systems, etc. Figure 7 on the next page shows a list of simulation participant objects.

The simulation activities and their associated objects are the basis for determining the size and scope of the battlespace. The activities may define the sequence of activities in a single engagement (such as the detection and destruction of an incoming missile), or for a single evolution (such as an underway replenishment), or for activities covering an ocean basin. By examining the objects associated with simulation activities the environment team have a basis to infer the environment parameters which affect platform motions, communications availability, sensor performance… the relevant operational behaviors.

As a result, the environment representation team understands the environment's spacial distribution and its spectral content.



Domain Objects
- WeaponRAM
- WeaponCIWS
- Threat
- Terrain
- Target
- Ship
- SeekerThreat
- Range Measurements
- RAMSeeker
- RAMMissile
- RAMLauncher
- RadarSPS_49
- Ocean
- EWSLQ_32
- Contact
- Command SSDS
- Clutter
- CIWSRadar
- CIWSGun
- Atmosphere
- PlaySpace

Environment System Engineer
Fed Simulation Engineer
PEO TSC System Engineer
ISD System Engineer
Combat System Engineer
Federate Developers
Environment Domain Experts
C:\WINDOWS\DESKTOP\Fall 98 SIW Pap
C:\WINDOWS\DESKTOP\Fall 98 SIW Brie
Logical View

For Help, press F1

Figure 7 List of Simulation Objects

2.1.3. Use Case View Step 3: Describe the Objects in Class Diagrams

The class diagram is used to describe the participant objects comprising the scenario to be simulated. We capture limited information about the object as a way of determining which environment parameters affect object behaviors in the scenario. The class diagrams answer the question "What participant characteristics are affected by environment?" With simpler scenarios it's often enough to simply record each of the participants as a class, and show relationships among the classes. For instance, a vehicle is a participant object that may be composed of a platform class, and one or more sensor and weapon classes. You might want to record mobility characteristics as operations of the platform. Or perhaps you would record environment parameters used in a combat direction system as attributes of the combat direction system class. The class diagram may contain the only reference to range instrumentation, capturing the range instrument behaviors as operations and the measured values as attributes.

There may be a substantial amount of participant information already captured in conceptual models of the mission space, or in federation or simulation object models. However, it's important to remember that the purpose of class diagrams in the use case is to record real world participants in the scenario. To the extent that conceptual or object model documentation abstracts or ignores the real world, duplicating that information in the use case only obfuscates your understanding. Leave representation decisions to the ECM's logical view. Figure 8 shows a use case class diagram.

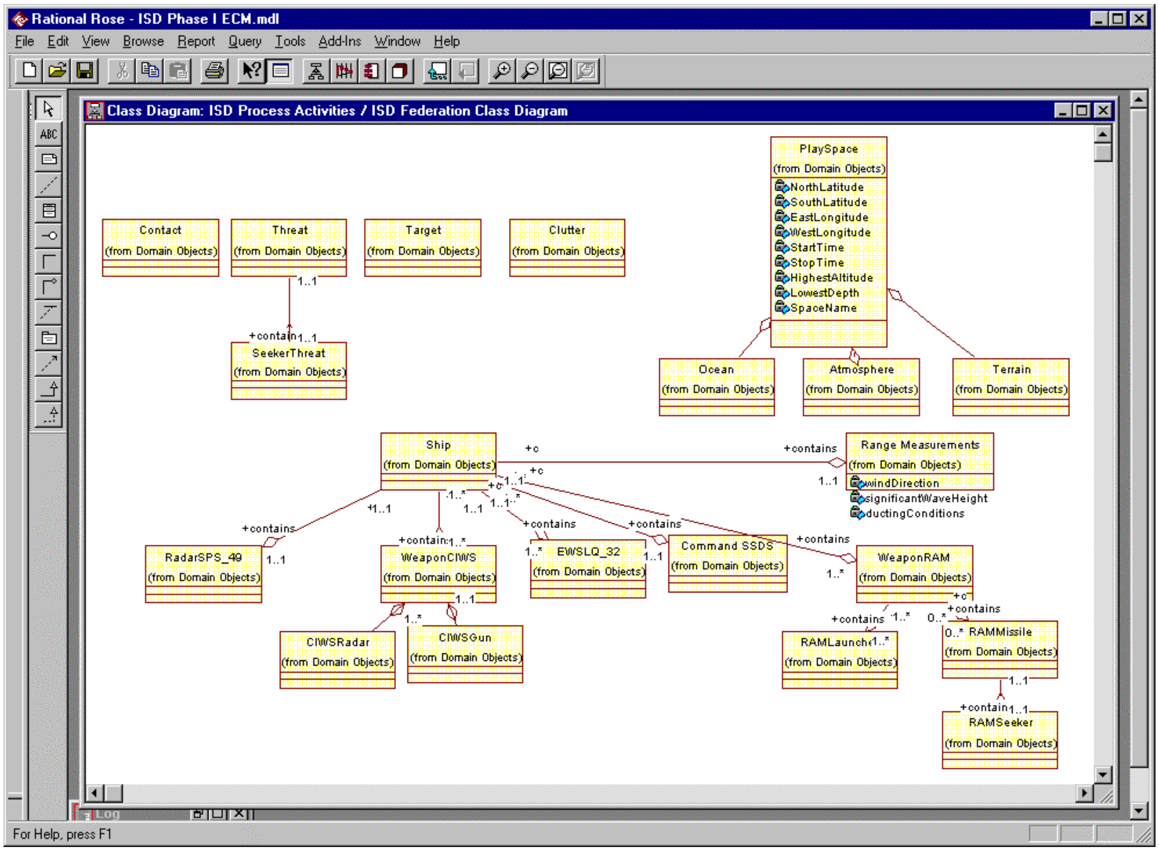


Figure 8 Military Objects in Use Case Class Diagram

2.1.4 Use Case View Step 4. Define Object Activity Timeline Sequence Diagram

The sequence diagram is used to represent the flow of activities over time. This diagram is particularly important for capturing the context of operational evolutions, where the sequence of activities or the temporal dimension is important. The sequence diagrams answer the question "How long is the simulation period?"

The sequence diagram helps to scope environment representation requirements by identifying the information exchanged between objects, and an order for exchanging that information. Generally, we only construct diagrams using objects that are environment-sensitive, and we only identify environment-sensitive information exchanges. (By environment-sensitive, we mean information such as contact reports, track updates, communications system service status reports; information which would be affected by environment conditions.) For simulations involving a single engagement the sequence diagram is often fairly simple. However, in more complex simulations, multiple diagrams may be needed.

We use the information in sequence diagrams to determine the duration of the scenarios, and thus the elapsed wall clock time in simulation space. If an engagement is completed in a few minutes, then it may sufficient to represent the environment statically, without considering such issues as transition from ducting to subrefraction conditions, or changing soil moisture levels. If a series of missions are simulated, then the environment may need to represent transitions from daylight to darkness, clear air to nil visibility, etc. Dynamic environments are more difficult to assemble, to control and to serve to simulations at runtime.

The complexity of multiple sequence diagrams may also give us a clue to ways to simplify the environment representation. If there is extensive interaction between a series of objects during one phase of a simulation, while other objects are "idle", then perhaps it is only necessary to be concerned with the environment representation in a portion of the battlespace at any one time during the simulation. If activities in the physical world are segmented, then perhaps segmented environment representations can be assembled and managed independently, preserving consistency at carefully determined segment boundaries. Simplified environment representations are easier to control and serve to simulations at runtime. Figure 9 shows the sequence diagram for the example single engagement simulation.

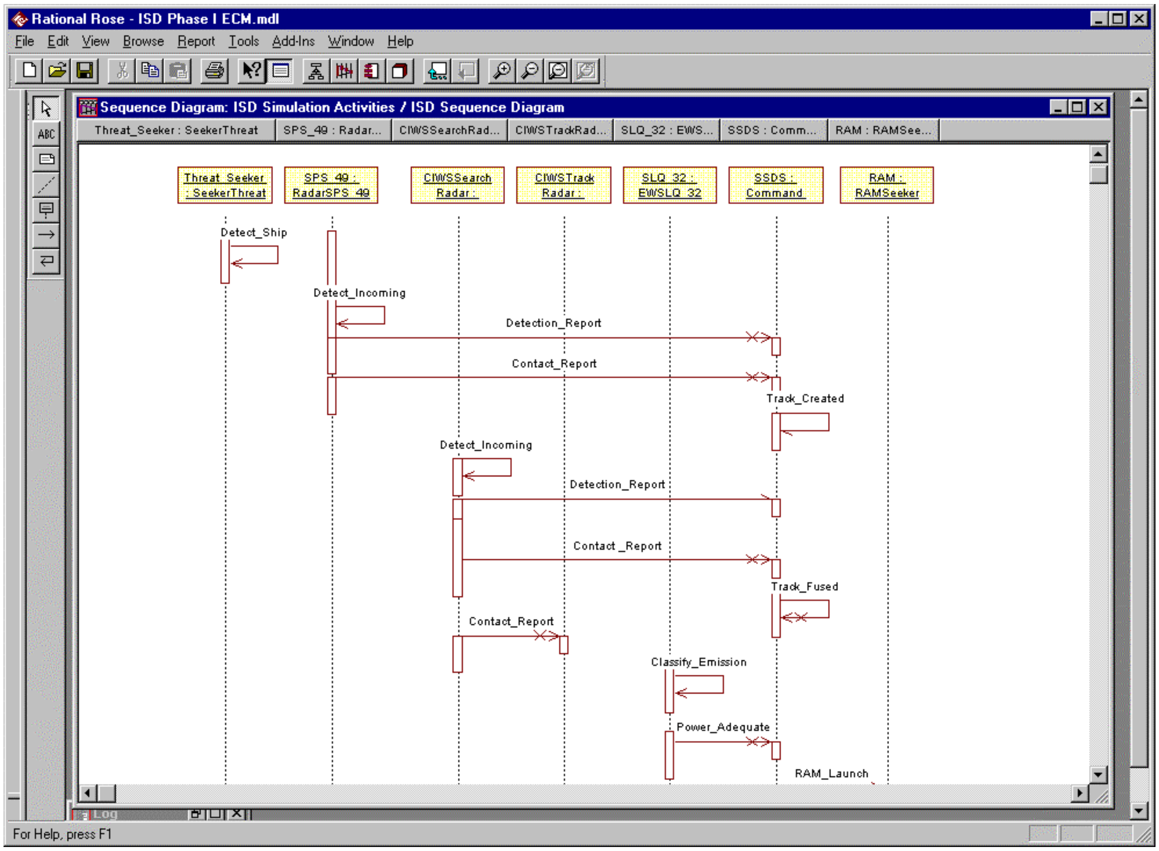


Figure 9 Use Case Sequence Diagram for Single Engagement

2.1.5 Use Case View Step 5 (Optional): Define Object Activity in a Context Diagram

However, we've found that the syntax of design languages (including UML), sometimes need a little help in communicating the essence of a tactical situation. In these cases we like to further mutual understanding by sketching a context diagram. A Context Diagram is simplified schematic of the tactical situation, much like a map display or a sandbox. These diagrams are particularly useful when verifying common understanding with combat system operators and environment domain experts, neither of whom may have a background in information systems design.

Context diagrams are simple schematics of a scenario. They record the spatial, temporal and spectral information. Often, a context diagram looks like annotated map or a sensor display. A context diagram is often a shorthand means of communicating with operational domain experts. You may be tempted to use a context diagram as the only means of capturing the context of a simulation. If the scenario is simple; and the simulation is simple; and there's no intent to enhance the simulation; and no intent to reuse the simulation; then a context diagram is all that's needed. But for the remaining 90% of simulation developments/applications, we think use case, sequence and class diagrams are a wise investment.

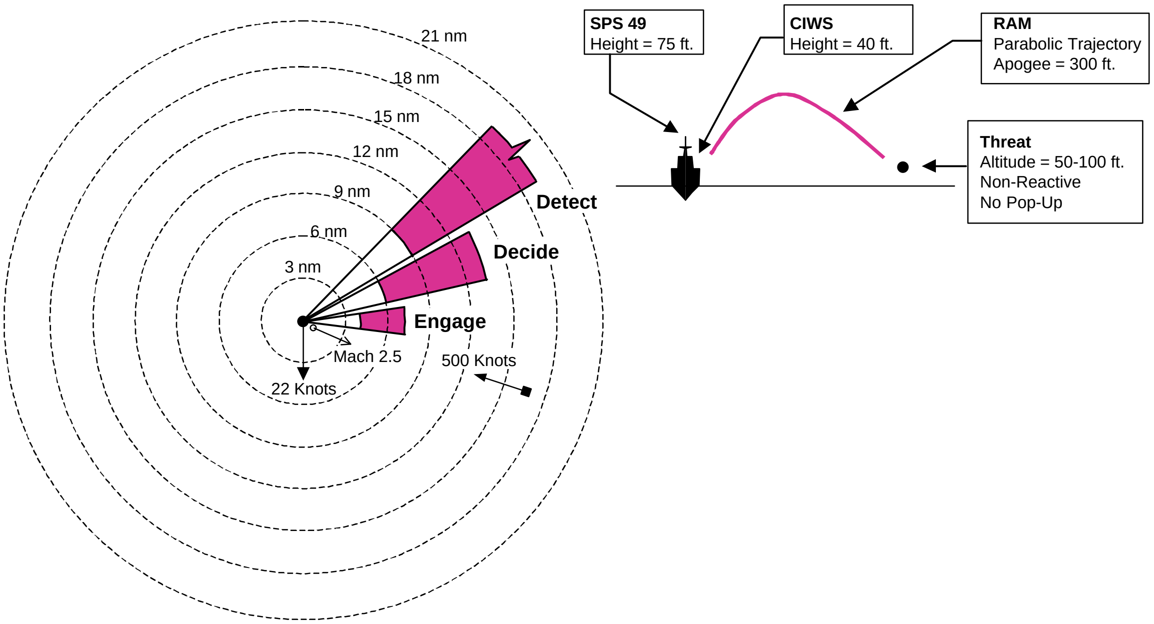


Figure 10 Context Diagram

2.2 Building the Inferred View

The inferred environment representation is the documentation of the environment that logically accompanies (supports) the battlespace and application context described in the use case. The UML representation diagram most often used in the inferred view are the class diagrams and interaction diagrams. The class diagrams are used to show the environment classes, and to record the relationship between the military objects in the use case and the environment classes. The interaction diagrams may be used when military objects create environment impacts (like wake or contrails) which must be integrated back into the consistent environment representation. (In this version of the document we have not included an example of an interaction diagram in the inferred view.)

The guiding principle for creating inferred representations is to capture the real, physical battlespace, to the extent that it affects the behaviors of the objects listed in the use case, for the application described in the use case. It's best to delay any other assumptions or simplifications until one begins to build the implemented view.

2.2.1 Inferred View Step 1: Develop/Download Stereotype Environment Class Structure

A stereotype environment class structure is a structured collection of attribute and operation descriptions for different environment domains. Normally, stereotype structures are hierarchical or relational, expressed as sub and superclasses, or as information structures with relation keys. The structures are stereotypical because they are used to organize a variety of databases, models, or object bases. The Master Environment Library (MEL) data dictionaries and the Synthetic Environment Data Repository Interchange Standards (SEDRIS) specifications are examples of environment stereotype data structures.

When building your first ECM, or for unique applications, you may wish to develop your own stereotype class structure. For broadly based representations, we recommend a structure with superclasses for terrain, ocean, atmosphere as well as a superclasses for constants and a superclass for coordinate notations. Within each such super class we would layer representation categories based on physical objects and processes, as appropriate. For instance a terrain superclass might have subclasses for topology, vegetation, cultural features, etc. (this object structure closely follows SEDRIS specifications.)

For more limited applications in a single domain, such as the Naval weapon system simulation in our continuing example, another approach may be easier to develop and use. In Figure 11, below, we have developed classes for various types of energy propagation phenomena such as scattering, specular reflection, dialectric values, and sea surface characterizations. We use this set of stereotypes because the environment-related processing in our example simulation is almost exclusively associated with radio frequency and infrared energy propagation and reflection.

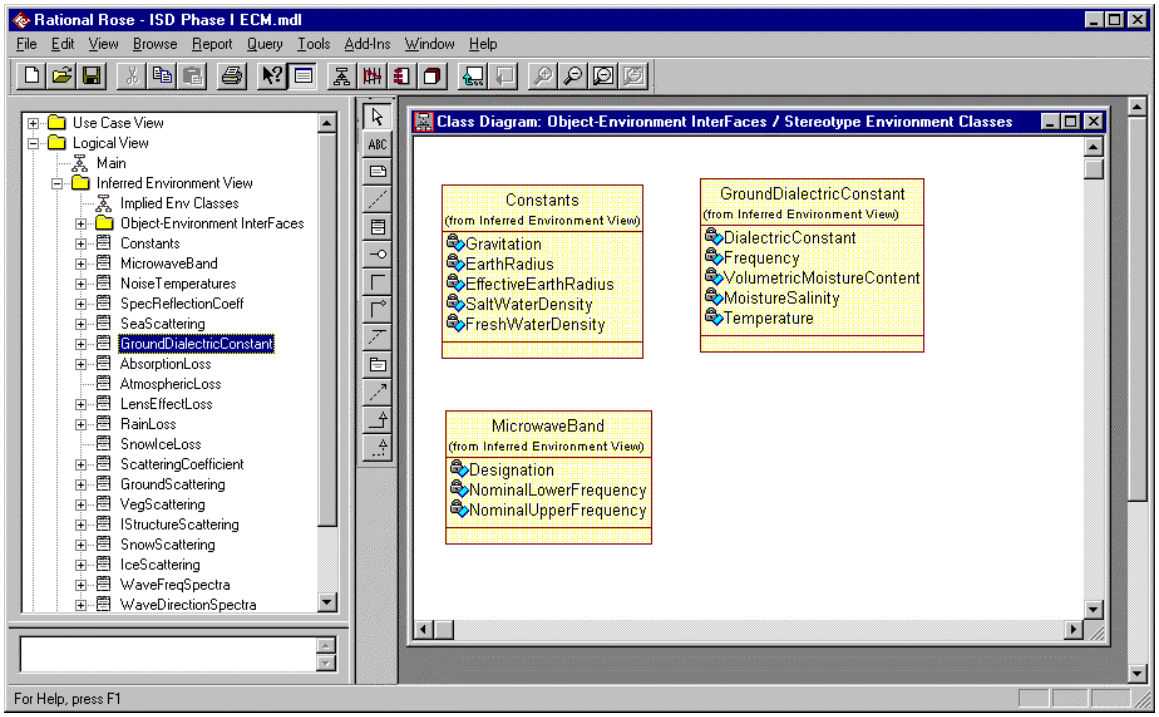


Figure 11 Example Stereotype Class Structures

Regardless of the scope of the class structure, our experience suggests that classes should always list attributes, but may or may not list operations. Operations (descriptors for algorithms or calculations) are often the determining factor in determining fidelity, and stereotypes are intended for a range of fidelities.

Whether the simulation application calls for a broadly based or limited environment representation, the stereotype class structure should include all the contemplated representation needs. This is especially true for stereotypes that are downloaded and reused. For stereotype class structures that are purpose-built, it is still useful to err on the side of excess. By using a stereotype class structure as the starting place for the inferred environment, we start by attempting to "covering all the bases". This approach allows the ECM documenters visibility into the extent of the needed environment representation at the earliest possible time. (Remember that one of the underlying motivations for the ECM is to provide feedback from the environment representation team to the simulation systems engineer early in the implementation process.)

2.2.2 Inferred View Step 2: Prune/Elaborate Stereotype Class Structure

With the stereotype class structure in place, the next step is to use the class structure as a baseline, auditing the baseline against the needs as described in the use case. The audit proceeds by inspecting the class structure, changing the class structure as needed to achieve the environment representation that reflects the scenario and participants of the use case. The changes might include changes to the class hierarchy, new classes, class deletions, and changes to the class attributes. Most importantly, the class structure may be elaborated by adding operations (descriptions of calculations) to the class descriptions. The operations represent possible effects and impacts calculations, and they play an important role in determining the level of fidelity of the environment representation.

It's tempting to view the audit process serially, first adjusting the classes for consistency, and then adjusting the classes for completeness. In practice, the process moves more quickly and easily if one iterates, alternately considering consistency and completeness issues. One resolves substantial issues in the early iterations, and considers more subtle implications of the use case in subsequent iterations. The following discussion considers consistency and completeness issues separately and serially, for clarity's sake. But apply the audit process iteratively. Figure 12 shows a more complete inferred environment class diagram.

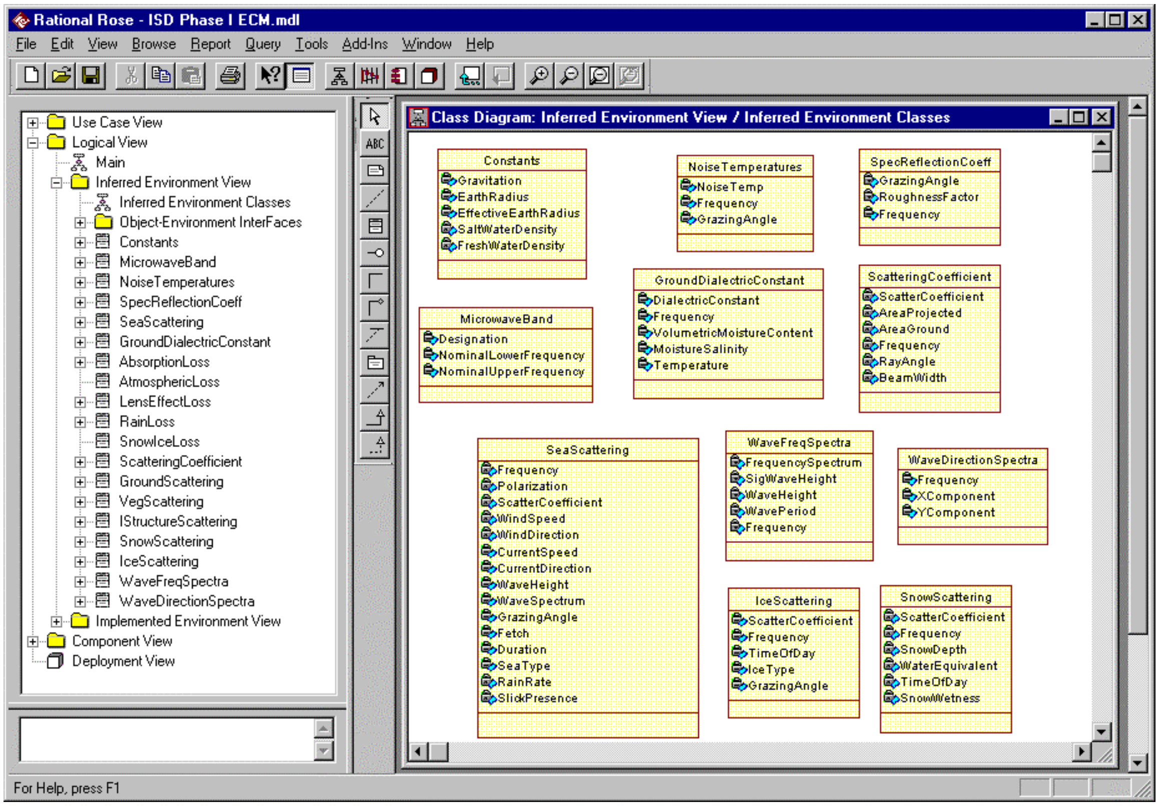


Figure 12 Inferred Environment Class Diagram

2.2.3 Inferred View Step 3: Check Inferred Classes for Consistency

There has been considerable debate about the precise meaning of "consistency" as the term applies to modeling and simulation. For the purposes of synthetic natural environment representation, we use a practical definition that serves our limited auditing purpose. The consistent synthetic natural environment:

a. Obeys the laws of nature and physics

b. Changes from state to state in a continuous manner, due to an underlying nature or physics-based causality
c. Replicates the real environment that it models.
d. Exhibits the above qualities at an agreed feature level of detail, time step, and length scale.
e. Exhibits an agreed smooth transition at any boundary between differing levels of fidelity.

The above definition suggests that there is no one consistent environment representation. Rather, consistent representations can be achieved for many combinations of parameters, both statically and dynamically, for many different levels of fidelity (or detail). Further, it's not necessary to enforce one level of fidelity throughout the simulation space, or for the every simulation time step[4].

Because there are so many possible consistent representations, our definition of consistency is more easily characterized by what it forbids than by what it includes. In the next few paragraphs we will offer some examples of representations that violate our definition of consistency.

There are several common sources of inconsistency in environment representations, and these sources fall roughly into 3 categories. First, there may be inconsistencies between the representations for difference environment regimes, (terrain, atmosphere, oceans, the surf zone , etc.), with respect to length scales, and boundary interfaces. Second, there may be inconsistencies between the time scale of simulation events and the time scale of the environment representation (static environment representations vice dynamic weather, etc.).. Third, there may be inconsistencies in the representation of environment effects at different bandwidths throughout the energy spectrum.

When the simulation space spans several environment regimes, the environment representation must often be drawn from several sources, representing the work of several different communities of domain experts. Often, each community has its own technical vocabulary, and its own standard practices. In the ECM inferred view, inspect each class for similar parameters or similar operations. For instance, if a terrain class uses topographic data, how does the tiling compare with the same topographic data in a bathymetry class? What is the exact meaning of the sound speed parameters for air and water? Difficulties occur most often at the boundaries, between regimes. The sea surface, and the coastal surf zone offer the starkest examples. At the sea surface, capillary wave amplitude and direction are often dependent on the surface wind strength and direction. At that boundary, aerosol content is quite different, and is the result of the interaction between wind and wave. In the surf zone there is stark contrast between soil density below the low water mark, above the high water mark, and in the tidal band between them. For highest fidelity, each might be modeled separately, with dynamic boundaries depending on the tidal cycle. It may be useful to employ a domain expert to review individual boundary assumptions, because of the specialized nature of many boundary interactions.

Time scale is a crucial issue to resolve early. Fundamentally, is the time scale of the simulation sufficiently short that environmental phenomena can be assumed to be static, or must environment be modeled as a non-deterministic, dynamic entity during the simulation? If the time scale permits static environment representations, or if environment dynamics are scripted, then data sets can be pre-assembled, environmental effects may be pre-calculated, and the resulting files distributed to the simulation components before runtime. However, if the environment is dynamic, and non-deterministic (influenced by unscripted simulation events) then environment data, effects models and runtime servers must all be responsive to the simulation event queue.

Spectral, or bandwidth-related inconsistencies often arise when simulating sensing and communications systems. Whenever the environment classes in the inferred view suggest that two different sensor or communications objects might use the same environment attribute or operation, Further, whenever

[4] While multi-resolution simulation is not common in training, analysis, or assessment applications, it is often used in high fidelity engineering applications. For instance, grid and mesh generators for fluid dynamics simulations often generate fine grids for regions of rapidly changing fluid pressure (near tips, leading and trailing edges, etc.) and use widely spaced grids in the uniform flow far field regions. This same approach is used in finite element analysis to accurately calculate local stress and shear flow at abrupt discontinuities in structures.

modeling sensors and/or communications systems that use different frequencies, any propagation, attenuation, scattering, background or clutter inferred environment classes should be inspected for frequency-dependant attributes and operations. This is another instance where it may be useful to employ en environment domain expert.

We have developed a question-and-answer checklist that helps to expose inconsistencies in inferred environment representations. This checklist is evolving; the current published version is provided as Appendix A.

2.2.4. Inferred View Step 4: Check Inferred Classes for Completeness (against process-related activities)

The complete inferred view provides "just enough" environment representation to satisfy the needs of the simulation. This means that simulation objects that are sensitive to environment change state when the causal environment state changes. Implicit in this definition of completeness is that there is a degree of state change that is agreed to be of operational significance.

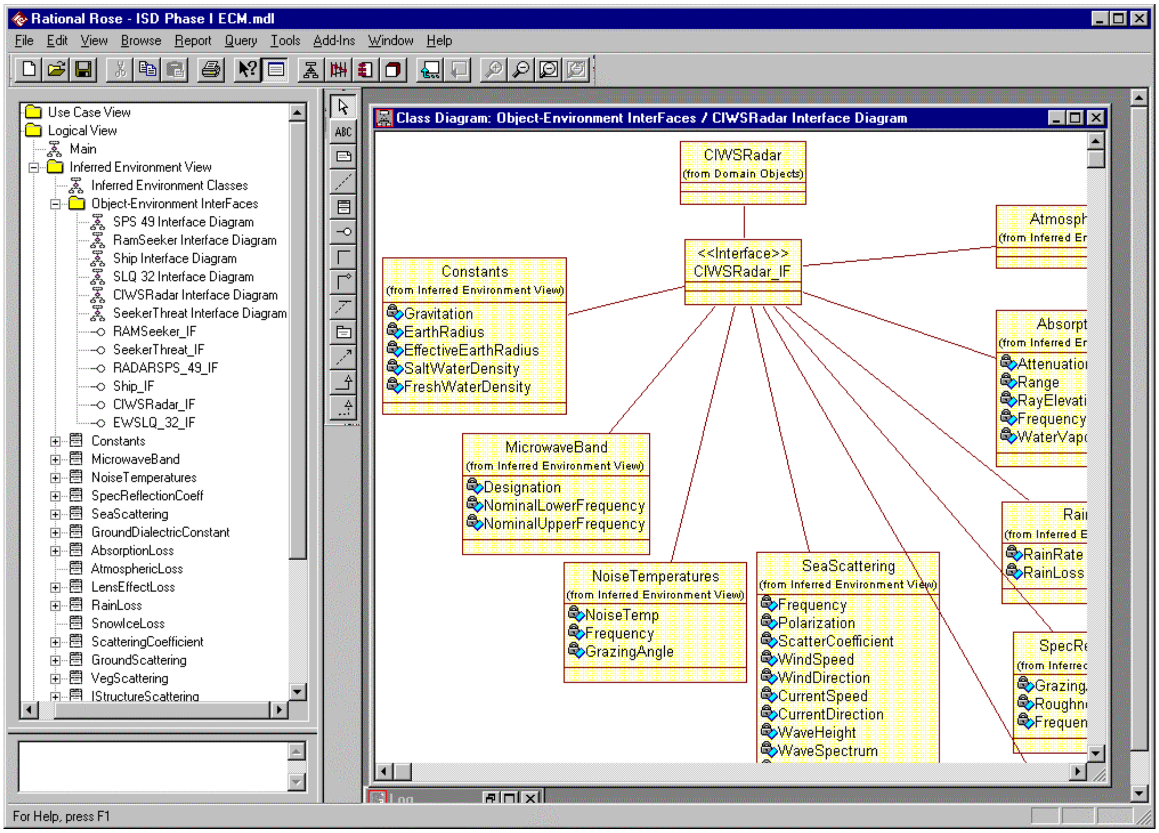
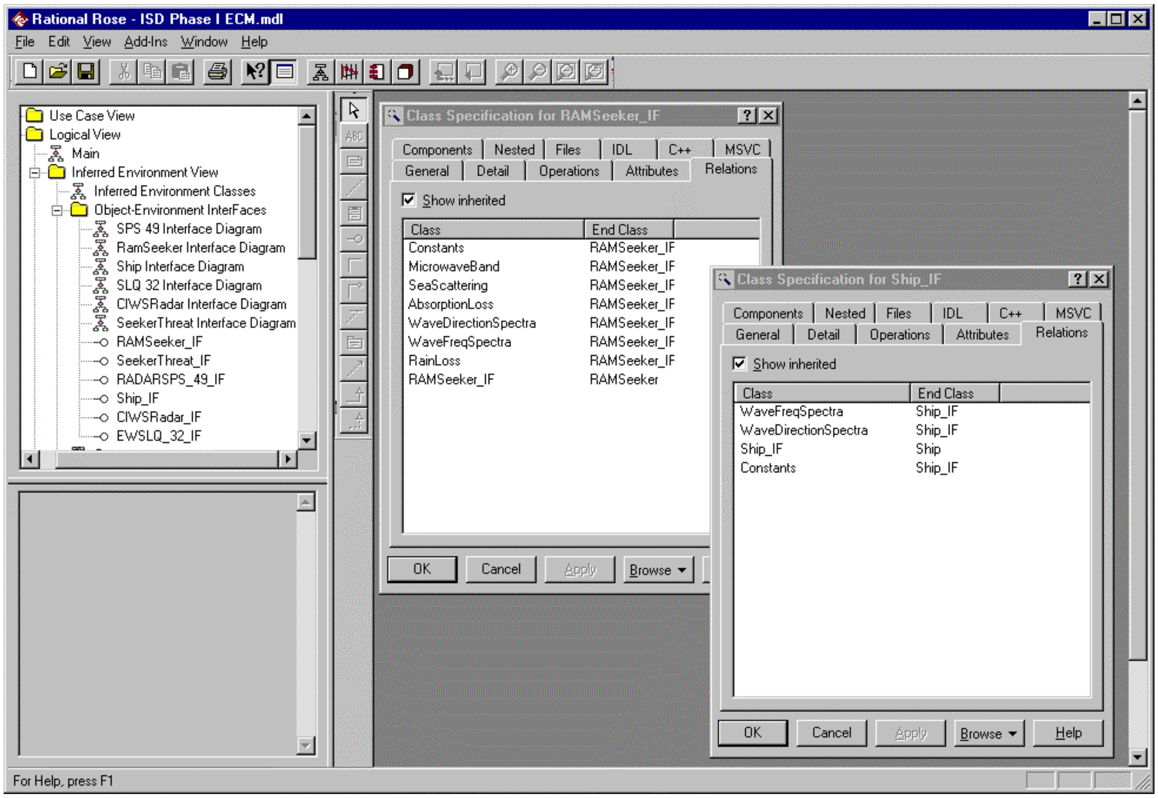


Figure 13 Class Diagram for Completeness Auditing

Thus the key to auditing the inferred environment view for completeness is to trace computational sequences from the simulation objects back through effects and impacts calculations, right through to environment data. There may be valid reasons why an object is not at all sensitive to environment. Often, however, this insensitivity is due to gaps in the environment data or calculation capability; the absence of environment classes, or missing attributes operations within a class.

We generally don't model the entire computational sequence in the ECM. Rather, we represent the interactions using class diagrams that employ interfaces[5]. Figure 13 shows a class diagram with three principal elements: A class representing objects (drawn from the objects listed in the use case), classes representing elements of the environment, (drawn from the inferred environment class diagram), and the interfaces. This simplified form frees us to concentrate on the goal of the audit; matching military simulation objects with environment classes. It's convenient to use one class diagram per interface, to avoid cluttering the diagrams.

As associations are created, we can see which object classes have no interface to an environment class, and also see if there are any "orphan" environment classes without an interface to another environment class or an object class. By opening the description for each object interface class, it is easy to inspect the relations list as shown in Figure 14 to determine which environment objects are associated with the object. Object


Rational Rose - ISD Phase I ECM.mdl
File Edit View Add-Ins Window Help
Use Case View
Logical View
Main
Inferred Environment View
Inferred Environment Classes
Object-Environment InterFaces
SPS 49 Interface Diagram
RamSeeker Interface Diagram
Ship Interface Diagram
SLQ 32 Interface Diagram
CIWSRadar Interface Diagram
SeekerThreat Interface Diagram
RAMSeeker_IF
SeekerThreat_IF
RADARSPS_49_IF
Ship_IF
CIWSRadar_IF
EWSLQ_32_IF
Components Nested Files IDL C++ MSVC
General Detail Operations Attributes Relations
OK Cancel Apply Browse Help
For Help, press F1

Class Specification for RAMSeeker_IF

☑ Show inherited

Class	End Class
Constants	RAMSeeker_IF
MicrowaveBand	RAMSeeker_IF
SeaScattering	RAMSeeker_IF
AbsorptionLoss	RAMSeeker_IF
WaveDirectionSpectra	RAMSeeker_IF
WaveFreqSpectra	RAMSeeker_IF
RainLoss	RAMSeeker_IF
RAMSeeker_IF	RAMSeeker

Class Specification for Ship_IF

☑ Show inherited

Class	End Class
WaveFreqSpectra	Ship_IF
WaveDirectionSpectra	Ship_IF
Ship_IF	Ship
Constants	Ship_IF

Figure 14 Interface Class Specifications

[5] In object oriented modeling, an interface describes the contract for a class or a component, without constraining its implementation. In UML, an interface can be documented in two ways. The first way, which is useful in the ECM, is to designate the interface using a "lollipop" symbol. This symbol is useful because we simply want to designate an interface between an environment class and a simulation object's behavior. The second way is to design the interface using an expanded form that resembles a class descriptor, without any attributes. This expanded form allows a modeler to visualize specifications inside the interface, and is useful for those who specifically wish to create interface specifications. Our example does not show the use of "lollipops" because Rational Rose does not support that convention. Instead, we have used the class descriptor without attributes.

classes without an environment class interface deserve further inspection to insure that there is no environment impact (at the desired fidelity). Similarly, one open the description for each environment class to inspect the relations tab and see that at least one object interface class uses the environment class. Orphan environment classes may well indicate environment "overkill".

The audit is complete when there are no unassociated environment classes, and there is a sound explanation for each unassociated behavior class.

2.3 Building the Implemented View

The implemented environment representation view is the documentation of the actual environment representation to be used in the simulation at runtime. The implemented view is, most often, the final compromise between the consistent, complete inferred environment representation and the realities of implementation budgets, schedules, proprietary software rights, available data sets, certified software, etc.

The representation diagrams used for the implemented view are the same as for the inferred view; class diagrams and interaction diagrams. However, in the implemented view, we associate components of the environment data and code with components of the simulation, if the implementation is composed of more than one component. Then, in each component we include the military objects to be simulated, relating them to the environment classes by using interface objects, just as in the inferred view.

Because the implemented view may include significant compromises in the environment representation, it may be necessary to document some of the assumptions inherent in the compromise decisions. Often, the best way to do this is by inserting comments in the classes. However, in the section on reverse engineering legacy representations, we will introduce a special type of environment attribute… the implied attribute.

The key differences in approach to developing the implemented view lie in whether the environment representation is being newly developed, or whether an existing representation is being modified for reuse. For newly developed environment representations the process proceeds forward from the inferred view, assembling the implemented view to accommodate the overall simulation implementation process. For reused/modified representations, the process is slightly more complex, because there is an existing implementation.

2.3.1 Implemented View for New Implementations

2.3.1.1 Implemented View for New Implementations Step 1: Choose the Environment Data and Calculations from the Inferred View and Prune

For a new implementation, the entire implementation team (simulation developers and environment domain experts) is proceeding from use-case-like descriptions of the simulation space. As a result, for new implementations, the inferred view is a good starting place for building the implemented view. The first version of the implemented view is thus created by reproducing the inferred view class and interface diagrams for the implemented view.

If this first implemented view is acceptable to the simulation system engineer, then the environment representation is substantially complete. Typically, however, the environment representation is subject to the same budget and schedule pressures as other components of the simulation implementation. As implementation compromises force changes in the simulation design, those compromises will probably drive compromises in the implementation view for the environment.

This short manual can't describe the many ways in which a synthetic natural environment representation can be revised. However, a few examples of compromises are listed below, with resulting change to the implemented view:

a. Fewer objects are used in the simulation; some environment classes may be eliminated
b. Fewer objects are environment-sensitive; some environment classes may be eliminated
c. A dynamic, changing environment can be approximated by several static environment representations; some effects and impacts calculations (class operations) can be eliminated
d. Fewer environment phenomena are represented; data and calculations (class attributes and operations) can be eliminated
e. Environment effects and impacts calculations can be simplified or approximated; calculations (class operations) are changed and data class attributes may be added.

In every case, there are changes to the implemented environment view, and they can be documented in the implemented view. As a result, the implemented view begins to diverge from the inferred view.

2.3.1.2 Implemented View for New Implementations Step 2: Associate Parameter Ranges and Models with Attributes and Operations of Classes in the Inferred View

As the simulation implementation proceeds, there is enough information available to more carefully specify the environment. The purpose of this further specification is to prepare data (attribute) and model (operations) requests from environment providers. The environment class attribute descriptions include initialization values, parameter ranges, and parameter increments. Environment class operations reference particular effects or impacts calculations, or comments may specify particular models. Again, the implemented view may continue to diverge from the consistency and completeness of the inferred view.

2.3.1.3 Implemented View for New Implementations Step 3: Compare the (Current) Implemented Representation with the Inferred View

Proceeding forward from the inferred to the implemented environment view is actually a collaborative process. The process involves the environment system engineer, environment domain experts, and simulation developers, according to the development process established by the simulation system engineer. In a waterfall development, one might expect to review the implemented environment view as part of simulation functional and critical design review. In a spiral development, the implemented view might be reviewed once design iteration.

At each review, the developing implemented view is compared with the previously prepared inferred view, and deviations from the inferred view are evaluated for their impact on environment representation consistency and completeness. Generally, deviations create local inconsistencies, with the result that there is a common consistent level of environment representation across the battlespace, with local areas, times and bandwidths for which the environment is inconsistent. At this point there is no general agreement on how to describe these inconsistencies; refer to section 2.3.4 for a description of the Consistent Environment Description document to see how inconsistency is described in our example.

2.3.2 Implemented View for Reused/Modified Implementations

2.3.2.1 Implemented View for Reused/Modified Implementations Step 1: Reverse Engineer the Existing Implementation and Associate Parameter Ranges and Models with Attributes and Operations

For reused/modified representations, there is an existing implementation. Therefore, as a first step, a data/model package is reverse engineered from documentation and code into the object model syntax. Figure 15 shows an implemented view that has been reverse engineered from existing simulation components.

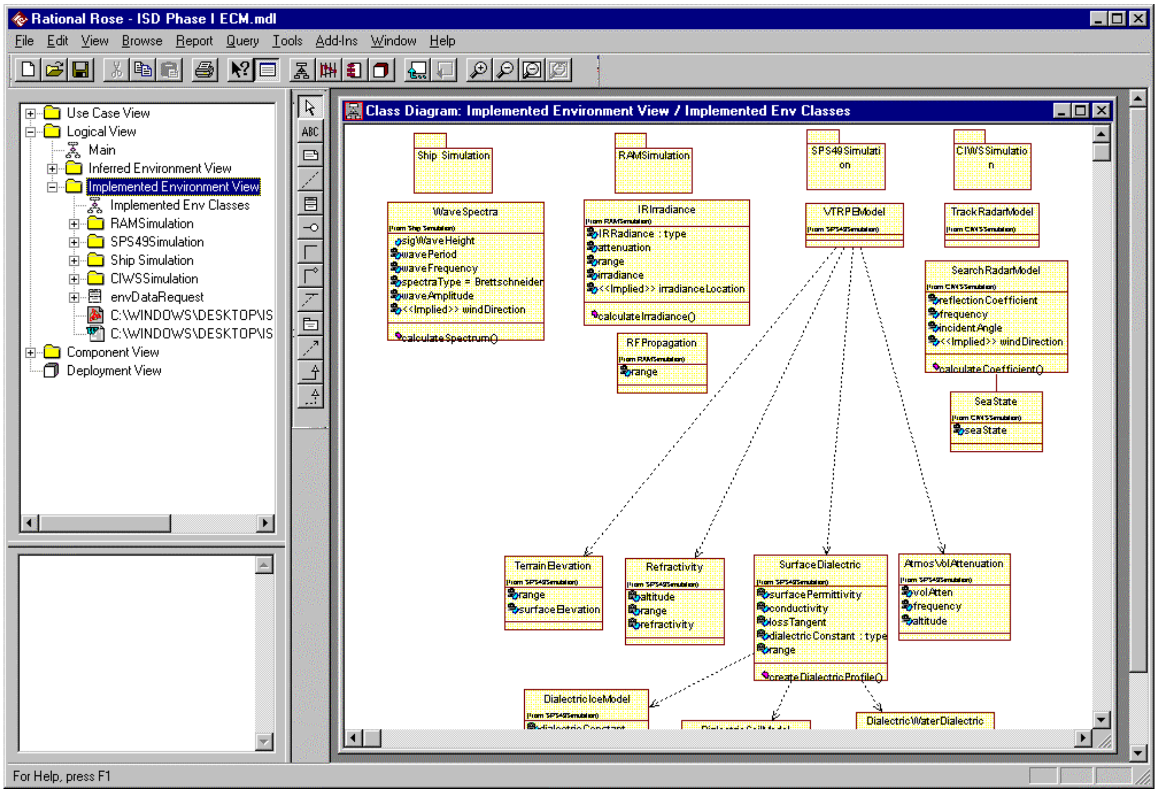


Figure 15 Reverse Engineered Implemented View

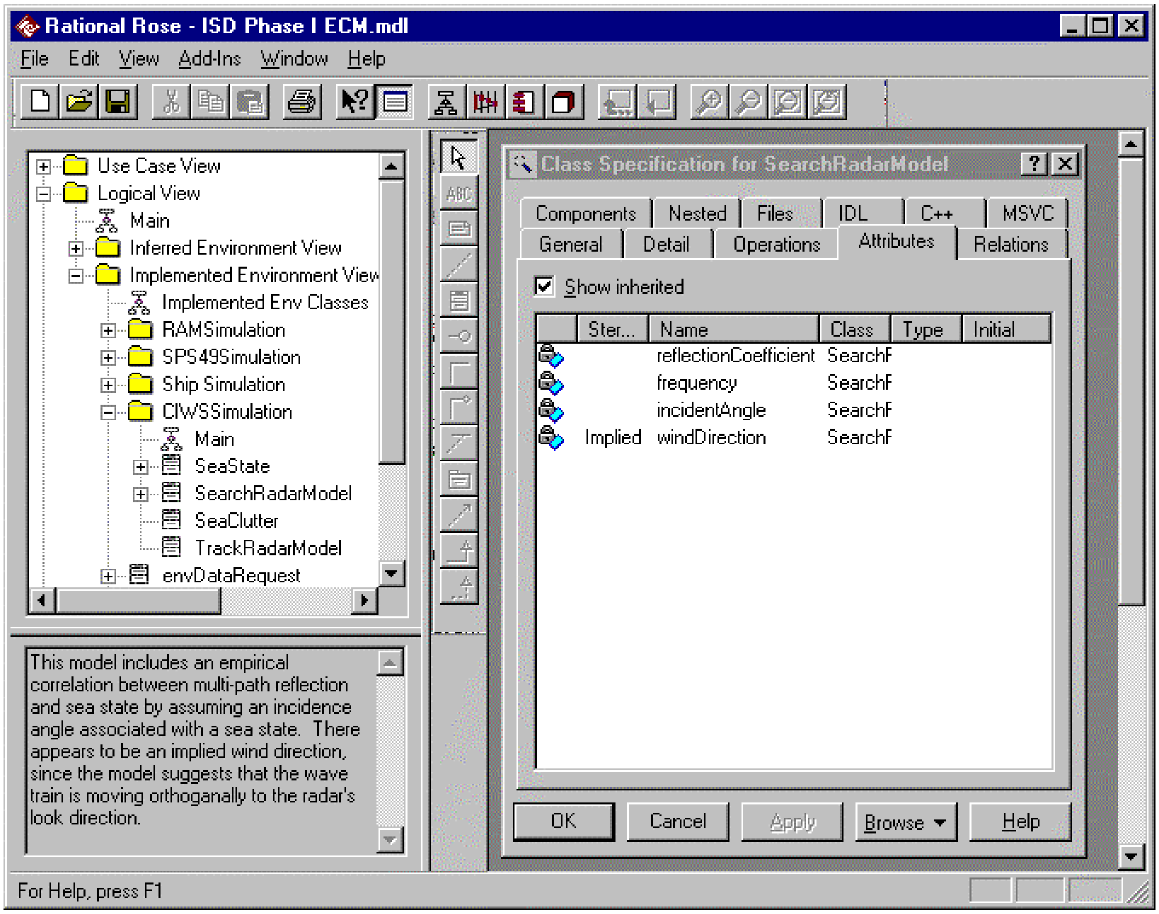
Again, we don't try to reverse engineer the entire simulation implementation; we limit ourselves to representing the components, the environment-sensitive military objects and behaviors, and the implemented environment representation.

2.3.2.2 Implemented View for Reused/Modified Implementations Step 2: Document Implicit Assumptions

In the example simulation the as-built simulation uses legacy environment calculations, embedded in components of the simulation code. For significant wave height, the algorithm assume a fixed wind direction relative to the ship's course. This assumption is deeply embedded in the calculation, but the calculation does not explicitly use wind direction as a parameter. To expose this assumption, we have chosen to add an attribute to the WaveSpectra class that services the Ship military object. The attribute,

WindDirection, is of a special type called "implied" which we create as an extension to the UML types[6]. The "implied" attribute does not physically exist in code, or in the algorithmic formulation of the calculation. By documenting assumptions using "implied" attributes, the effects of the assumption are very clearly exposed, and its impact can be tracked throughout the simulation. (We have also used the "implied" attribute type in the reflection coefficient calculation for the CIWS radar.) Figure 16 shows an example of a class that uses an "implied" attribute.

[6] UML allows the modeler to create new abstract data types, and we have exercised this capability to define a type uniquely suited to the needs of the ECM.


Rational Rose - ISD Phase I ECM.mdl
File Edit View Add-Ins Window Help
Use Case View
Logical View
Main
Inferred Environment View
Implemented Environment View
Implemented Env Classes
RAMSimulation
SPS49Simulation
Ship Simulation
CIWSSimulation
Main
SeaState
SearchRadarModel
SeaClutter
TrackRadarModel
envDataRequest
Class Specification for SearchRadarModel
Components Nested Files IDL C++ MSVC
General Detail Operations Attributes Relations
☑ Show inherited
OK Cancel Apply Browse Help
For Help, press F1

	Ster...	Name	Class	Type	Initial
		reflectionCoefficient	SearchF		
		frequency	SearchF		
		incidentAngle	SearchF		
	Implied	windDirection	SearchF		

This model includes an empirical correlation between multi-path reflection and sea state by assuming an incidence angle associated with a sea state. There appears to be an implied wind direction, since the model suggests that the wave train is moving orthoganally to the radar's look direction.

Figure 16 Class Using "Implied" Attribute

2.3.2.3 Implemented View for Reused/Modified Implementations Step 3: Compare the Reverse Engineered Representation with the Inferred View

Then it is possible to proceed as with a newly developed simulation, comparing the existing simulation with the inferred environment representation for each modification cycle. Again, the effect is probably to introduce local inconsistencies that must be identified and documented. At this point there is no general agreement on how to describe these inconsistencies; refer to section 2.3.4 for a description of the Consistent Environment Description document to see how inconsistency is described in our example.

2.3.3. Implemented View Step 4 (Optional): Describe Implementation in the Consistent Environment Description

As with use case documentation, it's often wise to offer an alternate description of the implemented view. For a use case alternative, we described the context diagram; as an alternative to the implemented view, we recommend the consistent environment description. The consistent environment description is a short summary of the implemented environment's impact on each simulation object, and any limitations on the object behavior validity caused by the scope of the implemented environment representation. Appendix B. shows an example of a consistent environment description. While the content of the description may differ, the subject headings and the nature of the discussions are wholly applicable to a wide range of environment representations.

A consistent environment description should include:

a. A summary which succinctly describes the common, consistent environment and any recommended changes to the simulation
b. A short description of the simulation scenario
c. A short description of the environment effect on each class or grouping of military objects
d. A statement of the overall impact of environment representation on the simulation
e. (Optional) a statement of environmental conditions which might have the most impact on military object behaviors
f. (Optional) any recommendations for modifying the simulation.

2.4. Implemented View Step 5: Prepare the ECM for Forwarding/Archiving

The ECM is documented in a standards-based notation, using standards-based tools. As a result, representations documented in the ECM can be exported for several purposes. The ECM use case, inferred view and implemented view can be exported to repositories to save as reusable representations, or to accompany reusable environment data and model sets. The ECM implemented view can also be used as input to object model template generators to quickly develop simulation object models or federation object models. Finally, the implemented view can be exported to application generators to generate database schemas or source code.

At this point the ECM content becomes implementation-specific, a part of the simulation software design and production process. Today, the particular export capabilities depend on which object oriented design and analysis tools is chosen to document the ECM. However, the software development trend is towards interoperable tools that can be chained to create an end-to-end software development environment.

If your organization or your projects already uses a suite of software engineering tools, it may be a worthwhile investment to choose a compatible ECM-building tool.

3. Maintaining the ECM

The ECM should be viewed as a living document, and maintained using the same mechanism as other project-related analysis, design and implementation documents. However, the nature of ECM content, frankly, complicates life cycle maintenance.

First, the ECM uses both static and dynamic object notations. Many existing repositories were originally intended to manage static database designs, and store only the static portions of object oriented models. Second, an ECM may include referenced files, and may also include information in alternate formats (use case context diagrams and implemented view consistent environment descriptions). As a result there is no simple way of archiving a complete ECM, and no simple way of globally modifying all ECM content from a single entry point.

Our present approach to maintain the ECM as a separate entity, uploading portions of the ECM to archives and repositories for specific purposes.

Appendix A. Environment Representation Inconsistencies Checklist

Appendix A

Consistency Question-and-Answer Checklist

General Questions

1. Over the time scale of the simulation, does the natural environment change (are there dynamic environment elements)?

2. If there are dynamic elements in the environment (waves, rain, wind, daylight/darkness, tides, seasons), what is the frequency of occurrence relative to the length of the scenario?

3. Do all simulation components use the same coordinate system? If not, is a translation utility provided?

4. Do all simulation components use a common definition for common parameters?

5. Do all effects calculations use the same values for physical constants for gravitation, standard temperatures, standard pressures, etc.? Is the definition of the constant uniform for every component? Is the number of significant figures used to express the constant appropriate to the agreed level of fidelity?

6. For environment effects, do all simulation components use the output of the same effects calculation? If not, do the various effects calculations produce the same result for the same input conditions? Is the difference to due an accounting for different phenomenology?

Ocean/Atmosphere Boundary Questions

1. Are the wave heights appropriate to the wind speed?

2. Is the period of the swell appropriate to the bathymetry?

3.Is the swell consistent with the fetch (distance of uninterrupted open water)

4. Is the direction of the swell appropriate to the prevailing wind direction?

5. For near-shore locations, does the local swell change due to coastline and bathymetry?

6. Is the presence of fog or mist appropriate to the difference in air and sea surface water temperatures?

7. Is the sea surface aerosol content appropriate to the incidences of breaking waves and surface salinity?

8. Is the ambient light level appropriate to the latitude, date and time of day?

Marine Vehicle Dynamics Questions

1. Is the modeled wave height and wave period within the vehicle's range of linear motion response?

2. Does the modeled wind speed affect the vehicle's motion response?

3. Is the vehicle sensitive to spray?

4. Does the vehicle's motion model use a stochastic or probabalistic wave model as input?

5. Does the vehicle's thermal model use sun angle as an input?

6. Does the vehicle's thermal model use ambient temperature as an input?

7. Does the vehicle create a "significant" impact on the environment?
    a. Wake
    b. Spray
    b. Exhaust plume
    c. Thermal scar

RF Sensor Questions

1. What is the difference in dB that that would cause a "significant" change in the behavior of the sensor being modeled?

2. How many bands must be modeled? Are the bands adjacent, or are the bands widely separated?

3. How is atmospheric attenuation accounted for?

4. Are ducting or subrefraction modeled?

5. What is the grazing angle of the propagated energy? Is the propagation model appropriate to the grazing angle?

6. Does the sensor's clutter process use a stochastic or probabalistic representation of clutter as input?

7. What types of clutter returns must be accounted for?

8. Are both gravity and capillary waves considered in the sea clutter model?

9. Are soil types considered in the land clutter model?

10. Is vegetation and vegetation type considered in the land clutter model?

IR Sensor Questions

1. What is the difference in irradiance that would cause a "significant" change in the behavior of the sensor being modeled?

2. What IR bands must be modeled?

3. What is the grazing angle of the propagated energy? Is the attenuation model appropriate to the grazing angle?

4. How is atmospheric attenuation accounted for?

5. Are particulate concentrations considered in the attenuation model?

6. Are humidity and rain considered in the attenuation model?

Appendix B. Example Consistent Environment Description

Appendix B

Integrated Ship Defense (ISD) Modeling and Simulation Pilot Program
Phase 1 Demonstration
Analysis of Environment Representation

Abstract and Summary

This paper analyses the individual environment representations embedded in the federates that constitute the ISD Phase 1 federation. The paper first reviews the geometry of the engagement scenario. Then the paper describes the scope and limitations of each federate's environment representation. Next, the paper describes the scope of the consistent environment parameters common to all environment representations. The ensuring section presents excursions from the common representations: unusual conditions that would unequally influence performance, a "least favorable" environment, and a "most favorable" environment. Finally, the paper provides two recommendations for simple modifications of the federate

The common, consistent environment is the "*least common denominator*" representation that is physically realistic:

> "A standard atmosphere prevails with median barometric pressures, moderate temperature, low to moderate humidity, no ducting conditions, wind speed of 10 to 15 knots, fully developed ocean waves of 3 to 5 ft (not influenced by adverse currents or shallow bathymetry), and the sun angle is below the horizon."

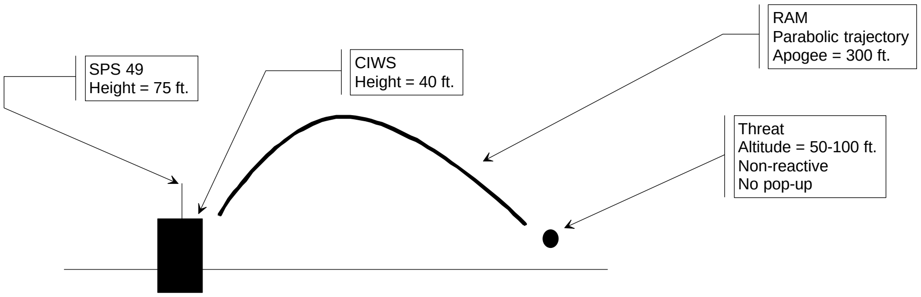
Although the consistent environment that results from the "*least common denominator*" representations is fairly restrictive, it is possible to selectively broaden the range of environments that can be represented by very selective ad hoc modifications.

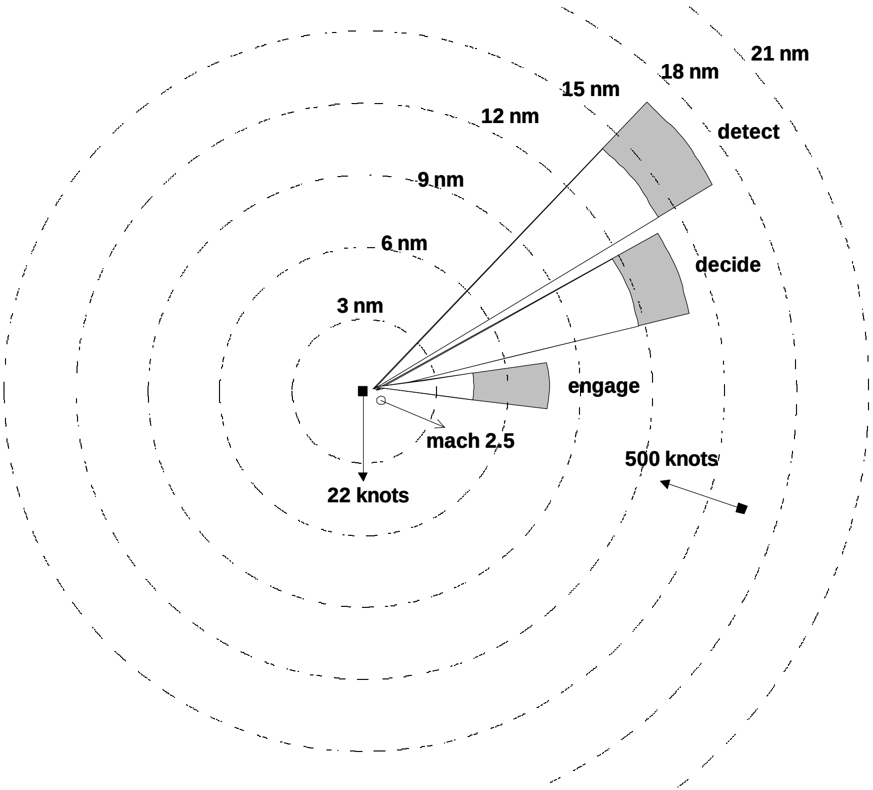
One may emulate the complete failure of a system detection due to severe environmental disturbance. This may be accomplished by inserting a software switch in each model that prevents sensors from reporting a detection.

One may emulate the change in detection range (due to ducting, subrefraction, rain, etc.) by inserting a scaling parameter in the range loss expression to arbitrarily increase or decrease detection range. The scaling parameter can also be applied to compensate for differences between calculated performance and operational test results.

Context of the of Phase 1 Scenario

The ISD mission space executes quickly, in a few minutes at most. Thus there is no need to attempt to represent dynamic weather patterns. The frequency bands of interest are the UHF, X, and KU bands in RF, and the medium wave IR band. The ship sensors and weapons and the threat missile(s) all operate near the sea surface where propagation effects are highly dependent on height of the sensor, altitude of the sea skimming target, grazing angle, ducting, subrefraction, horizontal fades, as well as shore and sea surface background clutter. This context information is important to establishing "just enough" environment representation. We found it useful to capture this information in a context diagram, showing the spatial, temporal and spectral elements of the scenario(s) of interest.





Assuming the geometries described above, then the minimum grazing angle is 0 degrees and the maximum grazing angle is about 1.6 to 2 degrees (depending upon the particular flight path of the missile after its apogee). Thus local surface effects are distributed over the entire flight path. Further, the grazing angle is never sufficient to move the geometry out of the range shallow angle regime. This general condition will continue through Phases 2 and 3, whenever the sea skimming threats are engaged using shipboard sensors and weapons.

Environment Effect on IR (RAM Block 0 Seeker only)

The provided irradiance data (for clear days in the North Sea and the Tropics) provide for a range of irradiance values which incorporate, by implication, the most important aspects of atmospheric attenuation over water: temperature and humidity.[i] In the geometries described above, clouds have only a very limited effect. In maritime scenarios, near the sea surface, the temperature can play a significant role as a cause of turbulence resulting in image blurring and scintillation.

High humidity has the effect of changing atmospheric attenuation. This might be most noticeable when comparing the IR seeker performance of marine tropical simulation scenarios (where humidity can easily reach 90%) with land-based test data (where the humidity might be as low as 20%). In general, the effect of humidity becomes more important at levels above 60%; one source indicates that "when the relative humidity increases from 75% to 95% near the ground, the horizontal visibility changes by a factor of two to six[ii]

The effect of humidity increases when there are high concentrations of particulate, around which the water vapor condenses. In maritime scenarios, near the sea surface, sea salt is the primary particulate[iii]. (The amount of spray, and therefore atmospheric sea salt, rises rapidly at and above Sea State 5 (wind speeds of 17-21 knots).

The provided irradiance curves described in the RAM Block 0 do not account for sun angle, lunar angle, abnormal humidity, fog, particulates, rain, or higher wind speeds/greater sea states. The effect of the sun is strongly dependent on the atmospheric attenuation and elevation angle of the sun. Small grazing angle,(sun low on the horizon), increases atmospheric absorption in the IR band thereby reducing the effect of the sun causing it to have a negligible influence.

Generally, fog, particulates and rain have the same effect of populating the atmosphere with particles that scatter radiant energy, increasing attenuation. Particle concentrations are highest near the earth's surface, dropping rapidly above 1 km. The most important elements of particulates are concentration and particle size. In maritime scenarios, the most important particulates are sea salt and smoke from damage.
The air/sea temperature differential plays an important role, since this differential is a primary indicator of the potential for surface fog, haze and general turbulence at the air/sea interface. (However, the observed data suggests that there is no direct correlation between these conditions and anomalous RF propagation. The report addresses correlation between microwave and mid-wave infrared propagation conditions. The RF band was 13.9 GHz Ku band; the IR band was in the 3 to 5 micron band. It appears that IR sensor range improves with positive differences in the air-sea temperature difference (ASTD). In these conditions, IR detection range varied from 10 NM with an ASTD of -3.25 degrees C to 16 miles with an ASTD of +2.12 degrees C. Note that microwave ducting or subrefraction can happen for both positive and negative ASTDs. [iv])

The provided irradiance model makes no provision for the variations in irradiance caused by scintillation effects. However, the seeker design is sufficiently simple to ignore scintillation effects, and thus there is no limitation in the simulation's performance caused by this simplification in environment representation.

Limiting conditions for Phase 1 environment validity using the existing RAM Block 0 Seeker IR model are as follows:

1. Model is not valid for time periods which would include dynamic environment parameters
2. Model is not valid for extreme range IR detections or when a low radiance target/threat is being modeled
3. Model is not valid for other than clear conditions
4. Model is not valid for wind speeds greater than Sea State 5 (17-21 knots)
5. Model is not valid for near-surface humidities above about 90%

It is possible to modestly extend the environment envelope by recalculating the curve fit parameters. Recalculations would allow for some different sensitivities to be explored via an environment parametric analysis.

Environment Effect on RF (SPS 49, CIWS, Ram Block 0 Seeker)

The three principal environmental effects that influence RF system performance are pattern propagation effects caused by the geometry and terrain, atmospheric attenuation, and scattering from clutter. For the ISD geometries, all three effects apply.

Rain. Attenuation from precipitation rises rapidly as the band center frequency rises, from 1.5 dB/km at 8 GHz (X band) to 14 dB/km at 14 GHz (Ku band) for heavy precipitation rates of 100 mm/hr. For another example, attenuation can vary from .01 dB/km for 1mm/hr rain rates to .37 dB/km for 16 mm/hr rain rates at 10 GHz. The attenuation due to rain is a highly statistical phenomena related to both time and location. Many factors effect the attenuation including drop size, increase in absolute humidity during the rain and the non linearity of rainfall rate vs. attenuation. This leads to a number of different models whose results can vary by as much as 3 db.

Sea Clutter. Generally, irregular surfaces such as the sea surface and land terrain are surface clutter problems, while the particle scatter problem (in the atmosphere) is a volume clutter problem. (Chaff is also a volume clutter problem). Each has a different law of variation, resulting in different range equations. The median wind speed over the world's oceans is about 15 knots, corresponding to sea state 3, which corresponds to a significant wave height of 3-5 feet, a duration of 15 hours, and a fetch of 100 nautical miles.

Sea surface clutter is predominantly represented as have a Rayleigh probability density function, but elements of the sea clutter have decidedly non-Rayleigh distributions. Gravity waves are generally thought of as having Rayleigh distributions, and are a function of wind speed (not necessarily the local wind), duration, and fetch. These are generally represented by the Pierson-Moscowitz spectrum, and can represent the underlying swell even when local wind is low. They are the primary drivers of large ship motions at sea. In shallow water, bathymetry increases the frequency and the wave height.

The local wind influences the capillary wave structure that overlays the gravity wave structure. This structure can cause Bragg scattering resulting in false targets due to random glinting and bright spots depending on the presence of gravity waves, relative direction of the local winds and currents.

The clutter coefficients vary with grazing angle and alignment to the direction of the wave train. At very low wind speeds, and for wind speeds above about 15 knots, it is not clear that wave characteristics (and therefore clutter) are directly related to wind speed. Since the dependence of sea clutter on wind speed is complex and uncertain, more parameters must be considered in any littoral scenario. It's also clear that the distribution used to characterize sea clutter depends on the grazing angle and, to some extent, the frequency of the radar.

Rain on calm water has the effect of inducing sea clutter; for instance, a rain rate of about 6 mm/hr has the effect of creating a clutter spectrum roughly comparable to a 10 knot wind, for low grazing angles.

Land Clutter. For land scatter, the relevant parameters are soil type, vegetation type, moisture level, snow type and depth, and ice depth. The value of constants in the clutter models for land clutter change for low (<10 degrees), intermediate (20-70 degrees) and high angles (>70 degrees), and for frequencies below or above about 6 GHz.

Models for backscatter from vegetation vary with season, moisture content, state of growth, and time of day.

SPS 49

Because of its long range, the SPS49 is most sensitive to pattern propagation effects. To more accurately account for the differences in range performance it is necessary to apply a fully wave equation approach to calculating pattern propagation factors for the low grazing angle geometries which are typical of the ISD geometries.

The described clutter modeling approach is static or quasistatic. It does not account for changes due to movement in the background, or changes in the scattering coefficients used. This is not a problem for Phase 1, where the scenario is minutes long. However, for longer scenarios that result in differences in the background (shipping, moving vehicles on land, changes in Sea State, etc.) the clutter models will need to be recomputed at run time.

From the material available it appears that the approach implies that a series of clutter and weather models are used to establish a background, comprised of terrain models, physical features, natural features, and sea surface. It is not clear that there is any process to do determine that the inputs used for these models are consistent.

The sea surface clutter model needs to be able to represent fully developed waves as well as the more complicated wave generation mechanisms that occur in the littoral regions.

Limiting conditions for Phase 1 environment validity using the existing SPS 49 model are as follows:

1. Clutter model is not valid for backgrounds in which the clutter is nonstationary
2. Sea Clutter model is partially valid for coastal seas for which the principal wave components are not fully developed gravity waves.

CIWS

The CIWS model includes provision for a sea clutter model. In the available documentation[v], the sea clutter model is described as matching distributions of "actual sea clutter." However, it is not clear what conditions are be modeled. In the absence of other data, it may be assumed that the distributions model the scattering of ocean waves in an average condition of Sea State 3. There is no provision for parametrically altering the clutter model. Consequently, clutter does not vary when changing azimuth relative to the wave train, or for rising seas or different wave types.

There is no provision for a land clutter model. Consequently, the model does not accommodate scenarios for land backgrounds within the horizon of the CIWS radars (10 or 15 miles, in clear conditions).

Atmospheric attenuation is neglected, since the R4 power loss calculation implies a free space assumption. Therefore, neither the search nor track radar representations are affected by evaporative ducts, horizontal fade zones, fog or precipitation.

Limiting conditions for Phase 1 environment validity using the existing CIWS search and track radar models are as follows:

1. Model is not valid for heavy precipitation or dense fog
2. Model is not valid for land cluttered backgrounds
3. Model is not valid for either ducting or subrefractive conditions
4. Model is not valid for sea conditions above about sea state 5
5. Model is not valid for near shore scenarios which include a land background

RAM Block 0 RF Seeker

The RAM RF seeker does not include a multipath calculation, neglects atmospheric attenuation, and does not consider background clutter. Therefore the missile seeker representation is not sensitive to evaporative ducts, horizontal fade zones, fog or precipitation.

Limiting conditions for Phase 1 environment validity using the existing RAM Block 0 Seeker RF model are as follows:

1. Model is not valid for use when exploring seeker response to stealthy or reactive threats

Overall Impact: the Common, Consistent Environment

Each of the embedded environment representation has individual limitations that must be honored across the federation environment representation. For the ISD Phase 1 Federation models, these limitations are often implicit, resulting from unstated assumptions driven by simplified environment models. The common, consistent environment is the "least common denominator" representation that is physically realistic:

> "A standard atmosphere prevails with median barometric pressures, moderate temperature, low to moderate humidity, no ducting conditions, wind speed of 10 to 15 knots, fully developed ocean waves of 3 to 5 ft (not influenced by adverse currents or shallow bathymetry), and the sun angle is below the horizon."

In addition, the resultant common, consistent environment imposes several restrictions on the scenario geometry. As a result, the scenario geometry has the following implied characteristics:

> "The platform is proceeding such that the target is upwind or downwind of the sensors/weapons. The geometry of the engagement is such that the target elevation as seen by the sensors remains less than approximately 10 degrees. The engagement effectiveness is not dependent upon low SNR detections or low confidence tracks."

Environment Excursions

Unusual conditions

For the chosen set of systems, certain environments will tend to exacerbate differences in the relative performance of system components. The examples provided below indicate environmental conditions which would unequally affect the performance of the ISD Phase 1 systems.

A local rainstorm will have dramatically shorten the detection range of both high frequency radars and medium wavelength IR sensors, without significantly affecting the performance of lower frequency search radars. In fact, if the rainstorm somewhat suppressed capillary waves in the sea surface, the result might be to somewhat improve the performance of low and medium frequency radars by reducing the sea surface clutter.

Coastal sea conditions might include swell from fully developed ocean waves that are compressed due to shallower water bathymetry. If there is a local current and local wind moving against the principal direction of the swell, seas will become short and steep with significant capillary waves superposed on the gravity wave train. This would have the effect of dramatically increasing sea clutter levels for all radars, without increasing the aerosol concentration that would further affect the performance of very high frequency radars and IR sensors.

Recent testing has shown that a highly negative air/sea temperature differential can reduce IR detection ranges up to 40%, and can occur simultaneously with ducting conditions that can double the range of a radar out to the radar horizon.

Least Favorable Condition

For the scenario and geometries of the ISD Phase 1 demonstration, the least favorable condition occurs, when the sun is overhead (least attenuation path, heating the water and causing a turbulence layer at the sea surface) when the ship is operating inshore in shoaling waters in heavy swell (the aftermath of a storm at sea) with wind against sea (10 knots or more,), the atmosphere is subrefractive and there is a strongly negative air/sea temperature differential. All sensors are dramatically affected.

Most favorable condition

For the scenario and geometries of the ISD Phase 1 demonstration, the most favorable condition occurs on a clear day with no swell or current, a wind of 5 to 10 knots, no ducting conditions and a positive air/sea temperature differential with low to moderate humidity.

Recommendations: Controlling Environmental Effects in the ISD Phase I Demonstration

Although the consistent environment that results from the representations is fairly restrictive, it is possible to selectively broaden the range of environments that can be represented by very selective ad hoc modifications as follows.

One may emulate the complete failure of a system detection due to severe environmental disturbance. This may be accomplished by inserting a software switch in each model that prevents sensors from reporting a detection.

One may emulate the change in detection range (due to ducting, subrefraction, rain, etc.) by inserting a scaling parameter in the range loss expression to arbitrarily increase or decrease detection range. The parameter can be initialized separately as part of preruntime data distribution to emulate the unequal impact of certain environment conditions, as described above. The scaling parameter can also be applied to compensate for differences between calculated performance and operational test results.

[i] RAM Block 0 Weapon System description provided by Allen Robins

[ii] Manual of Remote Sensing, Reeves , Robert (editor in chief) American Society of Photogrammetry, 1975, p 187.

[iii] Wolfe, W.L., Zissis, G.J., The Infrared Handbook, Environmental Research Institute of Michigan, 1993

[iv] Stapleton, et al., Correlation of Microwave and Mid-Wave Infrared Low Elevation Propagation Data, June 1996, NSWCDD/TR-95/60

[v] Phalanx Search and Acquisition HOL Ada Simulation Manual Version 1.0, 3 October 1994, TM 5G-PHX.017-1

Skolnick, M. Radar Handbook, McGraw-Hill, New York, NY, 1990